commodore Magazine

AÑO IV Núm. 36
Febrero 1987 300 Ptas.

REVISTA INDEPENDIENTE PARA USUARIOS

Joystick con microswitches

Los Biochips

Juegos de ingenio: Dentro del laberinto

SUPLEMENTO PROGRAMAS 12 PAGINAS

COMPUTACION A LA VELOCIDAD DE LA LUZ

Director:
Rubén Sanz
Redacción:
Teresa Aranda
Colaboradores:
José D. Arias
Alejandro de Mora-Losana
Paloma Saco
Diseño:
Benito Gil
Edita
PUBLINFORMATICA
Bravo Murillo, 377 - 5.º A
Telf.: 733 74 13. Madrid - 28020
Presidente:
Fernando Bolín
Director Editorial Revistas Usuarios:
Juan Arencibia
Director de ventas:
Antonio González
Jefe de Producción:
Miguel Onieva
Servicio al cliente:
Julia González - Telf.: 733 79 69
Coordinadora Publicidad:
Silvia Bolín
Publicidad Madrid:
Emilio García
Publicidad Barcelona:
María del Carmen Ríos
Pelayo, 12
Telf.: (93) 301 47 00
ext. 27-28 y (93) 318 02 89
08001 BARCELONA
Depósito Legal: M-6622-1984
Dirección, Redacción y Publicidad:
Bravo Murillo, 377 - 5.º A
Teléf.: 733 74 13
Distribuye: S.G.E.L.
Avda. Valdeparra, s/n.
Alcobendas. Madrid
Distribuidor en Venezuela:
SIPAM, S. A.
Avda. República Dominicana
Edif. FELTREE
Boleita Sur Caracas (Venezuela)
Distribuidora en Argentina:
Distribuidora Intercontinental
Sta. Magdalena, n.º 541
Buenos Aires (Argentina)

Fotocomposición: Consulgraf
Nicolás Morales, 34. 28019 Madrid
Fotomecánica: Karmat
Pantoja, 10. Madrid
Imprime: G. Velasco, S. A.
Esta publicación es miembro de la Asociación de Revistas de la Información, asociada a la Federación Internacional de Prensa Periódica FIPP.
El P.V.P. para Ceuta, Melilla y Canarias, incluido servicio aéreo, es de 300 ptas. sin IVA

Rogamos dirijan toda la correspondencia relacionada con suscripciones a Commodore Magazine. Edisa. Tel.: 415 97 12. López de Hoyos, 141-5. 28002 Madrid. Para todos los pagos reseñar solamente Commodore Magazine.
Para la compra de ejemplares atrasados dirigirse a la propia editorial Commodore Magazine, Bravo Murillo, 377 5º A. Tel.: 733 74 13. 28020 Madrid.

Editorial

El hombre inventó el ordenador, al igual que otras máquinas, para que realizase algunas tareas rápidamente, consiguiendo en el menor tiempo posible unos resultados más exactos.

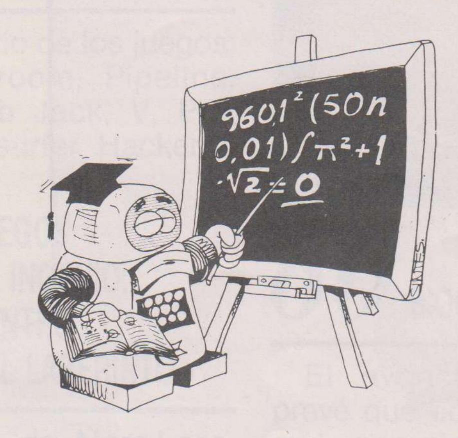
Una de estas razones que hacen del ordenador un objeto superior es su capacidad de cálculo. Extensas y complicadas operaciones pueden ser realizadas a tal velocidad, que un hombre en toda su vida hubiese sido incapaz de terminar.

Diariamente, miles de ordenadores realizan infinitas operaciones para controlar y establecer la economía de nuestro país, o para calcular la seguridad del material de los aviones, trenes o barcos que nos transportan.

¿Por qué no puede ser nuestro Commodore el artífice de estos insuperables logros? Este mes hemos decidido hacer que nuestro ordenador se convierta en un matemático exacto, sin desperdiciar ningún décimal ni redondear ninguna cifra.

Y es que como alguien dijo: «Nos enamoramos de las máquinas porque son los únicos seres lógicos que responden a nuestras órdenes con acciones exactas y sin posibilidad de error.»

Enamorémonos de nuestro Commodore, dándonos cuenta de lo perfecto y capaz, al calcular cualquiera de nuestros caprichos matemáticos, pues a quién no se le ha plantado alguna vez averiguar cuántos ordenadores hay en el mundo o cuántos granos de arena hay en la playa.



Sumar


rio



NOTICIAS

4.ª Fira de la Informática a Sabadell

Se llevará a cabo entre los días 2 y 8 del próximo mes de febrero, en Sabadell.

La JOVEN CAMARA DE SABADELL es una Entidad sin finalidad de lucro, apolítica y aconfesional, que tiene por objetivo primordial la autosuperación y formación personal, que se consigue con la realización práctica de programas que puedan tener una incidencia beneficiosa para la comunidad, y promuevan el desarrollo cultural, social y económico.

A lo largo de esta Feria se pretende difundir la informática en general, como tradicionalmente ya se ha hecho en las anteriores ediciones, y dar soporte este año, con especial relevancia, a la Robótica y a las Nuevas Tecnologías.

Paralelamente a la celebración de la Feria tendrán lugar otras actividades:

— Libramiento de premios del 2.° Concurso de Software CIUTAT DE SABADELL, organizado por la misma Entidad, en el cual pueden participar escolares y profesionales.

— Jornadas Técnicas, en las cuales se impartirán conferencias y seminarios sobre temas de vanguardia del mundo informático y las nuevas tecnologías.

— Programa para escolares que visitan la Feria de manera organizada.

II Concurso de software Ciutat de Sabadell

Alfons XII, 47
Telf.: 725 49 11
SABADELL (España)



Específico para escolares:

—Tema A: MASCOTA JUEGOS OLIMPICOS BARCELONA-92.

—Tema B: LIBRE

BASES

Art. 1: Podrán presentarse un máximo de 5 trabajos por escuela (previamente seleccionados por ésta).

Art. 2: Los programas se habrán de presentar obligatoriamente con:

— Identificación de la escuela.

— Nivel de estudios.

— Nombre y apellidos de los concursantes.

— Nombre del programa (máx. 8 caracteres).

— Soporte en cinta o disquet por duplicado.

— Indicación del modelo de ordenador, de la configuración necesaria, del sistema operativo (MS-DOS, CP/M o MSX) y del lenguaje utilizado.

— Opcionalmente se podrá presentar un listado del programa.

Art. 3: Los trabajos habrán de presentarse antes del día 25 de enero de 1987.

La decisión del Jurado se hará pública el día 7 de febrero de 1987, y el libramiento de premios el día 8 del mismo mes. La presentación de los programas se hará en la secretaría de la 4.ª FERIA DE LA INFORMATICA A SABADELL, situada en el edificio de la CAMARA DE COMERCIO DE SABADELL (c./ Alfonso XIII, n.° 43) por cualquiera de los medios usuales de tramitación, por correo o bien personalmente.

Art. 4: El Jurado estará formado por miembros de la JOVEN CAMARA DE SABADELL y por especialistas en informática.

Art. 5: La decisión del Jurado será inapelable y por tanto no se podrán presentar recursos.

Art. 6: La JOVEN CAMARA DE SABADELL se reserva los derechos de publicación de los trabajos premiados en el concurso, así como declarar desierto el premio por falta de calidad en los programas concursantes.

Art. 7: Se otorgará como primer premio un ordenador personal. El Jurado podrá otorgar también hasta dos accesits a los concursantes que no hayan obtenido el primer premio.

Art. 8: El hecho de participar en el concurso implica la aceptación de todas y cada una de las bases.

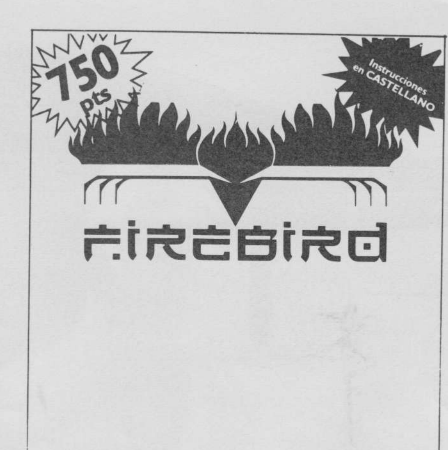
Exclusiva para Drosoft

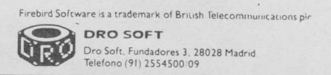
Estimados amigos:

Queremos comunicaros la firma por parte de DRO SOFT del contrato para la distribución en exclusiva de la serie SILVER (750 ptas. P.V.P.) de FIREBIRD. La presentación de dicho producto es la siguiente:

1.° PRODUCTO EMBLISTADO.
2.° INSTRUCCIONES EN CASTELLANO.
3.° P.V.P. MARCADO SOBRE LOS MARGENES DEL BLISTER.

Tanto de MASTERTRONIC como de las series SILVER (FIREBIRD) SOMOS LOS LICENCIATARIOS EN EXCLUSIVA PARA ESPAÑA y podemos demostrarlo con los contratos firmados en la mano.



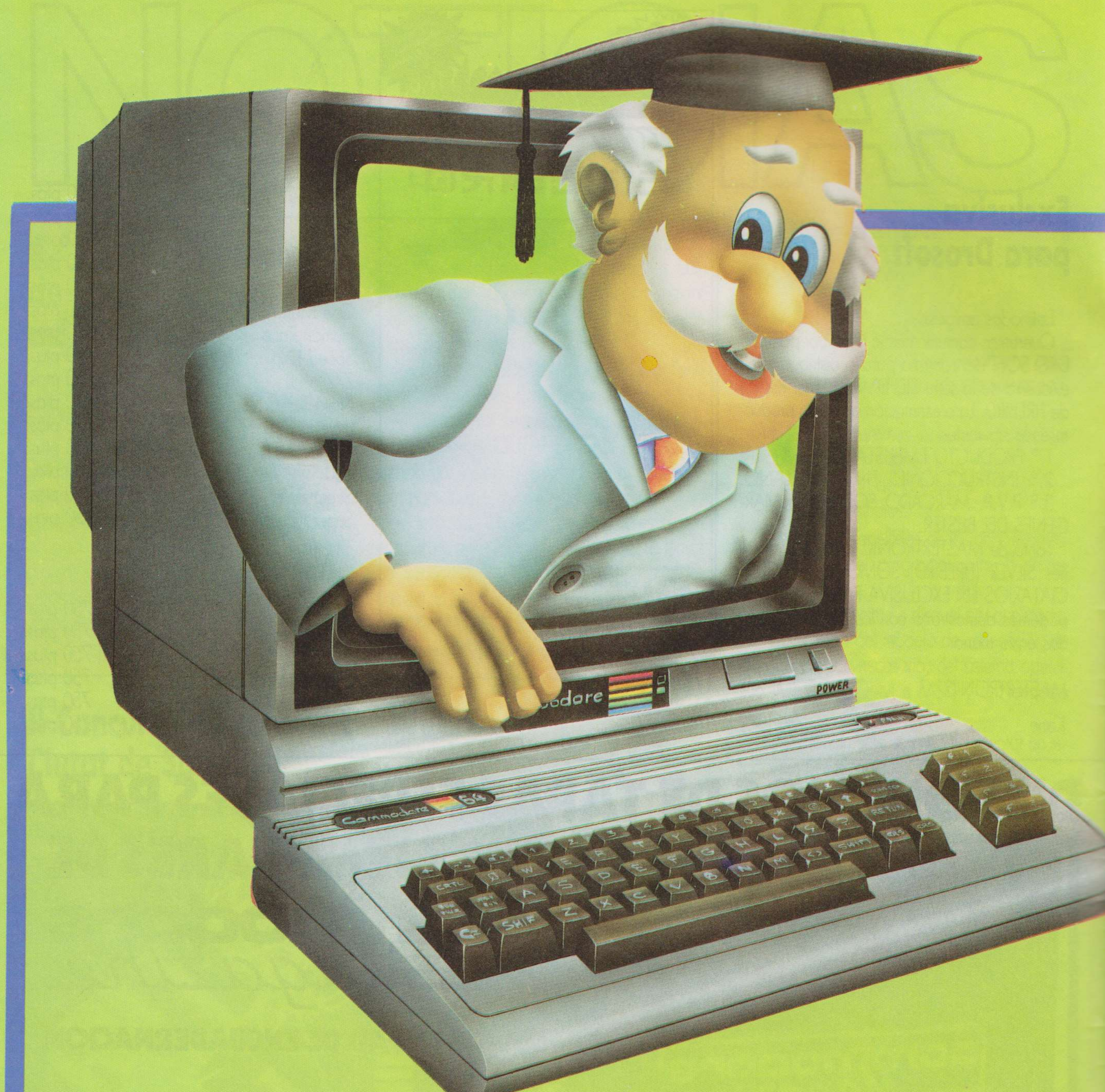


MASTERTRONIC

Kane	750 ptas.
Action Biker	750 ptas.
Finders Keepers	750 ptas.
Nonterraqueus	750 ptas.
Five a Side	1.100 ptas.
1985 the Day After	750 ptas.
Talisman	1.100 ptas.
Last V8	1.100 ptas.
Bump Set	750 ptas.
Kik Start	750 ptas.
One Man And His Droid	750 ptas.
Human Race	750 ptas.
Speed King	750 ptas.
ZZZ	750 ptas.
Ninja	1.100 ptas.
Hole In One	1.100 ptas.
Hollywood	750 ptas.
180	1.100 ptas.
Street Surfer	750 ptas.
Super Pipeline II	750 ptas.

SILVER FIREBIRD

Thrust	750 ptas.
Spiky Harold	750 ptas.
Ninja Master	750 ptas.
Harvey Headbanger	750 ptas.
Mad Nurse	750 ptas.
Warhawk	750 ptas.



Computación a la velocidad de la luz

Ahora tienes la oportunidad de introducirte en el mundo de los números con la misma facilidad que los ingenieros de la NASA. A través de unas pequeñas subrutinas que intentarán simular al superordenador Cray I a la búsqueda de los números primos desconocidos.

En Europa existe un único Cray-I, el superordenador más potente que jamás se ha construido. Lo tiene el instituto Meteorológico Inglés, y a él están conectados muchos de los demás institutos meteorológicos, como el español (que tiene derecho a una hora de su CPU). Basta con conectarse, enviarle el programa y los resultados son instantáneos. En Canadá existe otro, y en Estados Unidos la McDonell Douglas, una de las multinacionales más grandes de la aeronáutica, posee uno más. Su valor es de unos 3.000 millones de pesetas y sólo para cosas muy especializadas, como el estudio de los fluidos, pueden requerir ordenadores así.

La NASA americana tiene 12. Son el núcleo de un sistema mucho más complejo. Los terminales sobre los que se introducen los programas van a miniordenadores, y son éstos los que hacen mucha parte del trabajo, y sólo cuando es un trabajo demasiado grande, lo envían a unos «Mainframes», o grandes ordenadores, de IBM, con una potencia muy superior a la de los miles de minis conectados en todo el país.

Realizan el trabajo a una velocidad vertiginosa, y lo devuelven a los minis sin que apenas nos hallamos dado cuenta de lo sucedido. Sólo cuando el trabajo es excesivo, cuando son demasiados los cálculos incluso para el main-frame, pasa el trabajo a uno de los monstruosos cerebros Cray-I, siendo él el encargado de esos trabajos que requieren ingentes masas de operaciones. Nadie puede tratar directamente con él, salvo casos muy excepcionales. Precisamente de uno de esos trabajos excepcionales vamos a hablar.

NUMEROS PRIMOS

Los números primos se caracterizan porque son divisibles únicamente entre sí mismo y entre la unidad. Aparecen de una forma muy extraña, sin una fórmula real. El primero es el 1, después el 2, el 3, el 5, el 7, y cada vez se van distanciando más, pero sin una regla fija. Muchos matemáticos han intentado conseguir una fórmula general, pero nadie lo ha conseguido. Sólo existen algunas fórmulas aproximadas, y una cierta regla que dice: «si un número **n** es primo, existen grandes posibilidades de que $2^n - 1$ también lo sea».

¿Y cuál es el mayor número primo? ¿Existe un fin, un último número primo, o son infinitos?

Encontrar el mayor número primo se convirtió en una especie de reto para los matemáticos, sobre todo para aquellos relacionados de alguna forma con los ordenadores, ya que sólo ellos podían operar con números tan gigantescos.

Mervin Mersenne, un matemático y teólogo del siglo XVIII vaticinó que, siendo el **251** un número primo, $2^{251} - 1$ también lo fuese. Este número era una cifra muy grande como para poder calcular a mano sus factores, y hasta mucho tiempo después nadie pudo comprobar si realmente era primo o no y, a no ser por los ordenadores, nunca lo hubiésemos sabido. Pero técnicos especiales de la NASA pusieron manos a la obra y diseñaron un programa especial para un Cray-I, que dijera a la humanidad la respuesta al enigma. (Curiosamente, $2^{251} - 1$ en binario se obtiene colocando 251 unos seguidos, lo que ocuparía un total de 251/8 = 32 *bytes*).

EL ENIGMA

Después de más de 32 horas de durísimo trabajo, el fantástico Cray-1 descubrió el enigma. El número de ¡69 dígitos!, resultado de restar uno, a dos elevado a la 251 potencia (2^{251} — 1), no era un número primo, pues era divisible entre tres factores primos.

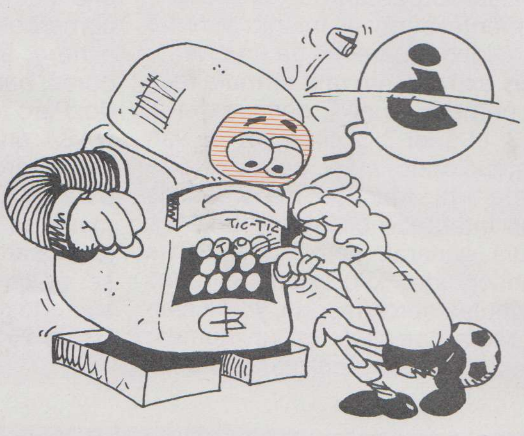
Ahora nosotros vamos a intentar emular de alguna manera al poderoso Cray-1, este mes quitaremos las telarañas a la circuitería de cálculo de nuestro Commodore para hacerlo trabajar.

La velocidad de un ordenador se mide en «flops», que son las operaciones de coma-flotante por segundo que podemos realizar. En microordenadores como el nuestro oscila alrededor de varios cientos (es decir, cientos de multiplicaciones, divisiones, etc... con números reales, o, lo que es lo mismo, números con decimales, todo lo complicados o largos que podamos imaginarnos).

Si lo comparamos con nuestra velocidad, tiempo que nosotros emplearíamos en realizarlo manualmente, no llega a una sola operación por segundo (alrededor de 0.1 flops para una persona muy rápida), los micros son auténticos genios en los cálculos. Los superordenadores Cray-1 o Cyber son muy rápidos. Su velocidad es también de cientos... pero de millones de flops, o mega-flops. ¡Millones de multiplicaciones en un segundo! Realmente, en un mundo en que nada nos asombra, es posible que pase desapercibido lo que esta cantidad significa, pero poniéndose a pensar lo que son millones de divisio-

```
9 REM *** NUMEROS ***
10 INPUT"PRIMER NUMERO A MULTIPLICAR";A$
20 INPUT"SEGUNDO NUMERO A MULTIPLICAR";B$
99 REM *** INICIALIZACION ***
100 LA=LEN(A$): LB=LEN(B$)
110 DIM A(LA),B(LB),D(LA+LB)
120 DEF FNU(X)=X-10*INT(X/10)
130 DEF FNC(X)=INT(X/10)
140 DEF FNM3(X)=ABS(X)-3*INT(ABS(X)/3)
199 REM *** MATRICES ***
200 FOR N=1 TO LA
210 A(N)=VAL(MID$(A$,LA-N+1,1))
220 NEXT
230 FOR N=1 TO LB
240 B(N)=VAL(MID$(B$,LB-N+1,1))
250 NEXT
299 REM *** MULTIPLICACION ***
300 FOR N=1 TO LA+LB: D(N)=0: NEXT
310 FOR N=1 TO LB
320 FOR M=1 TO LA
330 MN=M+N-1
340 T=D(MN)+A(M)*B(N)
350 D(MN)=FNU(T)
360 D(MN+1)=D(MN+1)+FNC(T)
370 NEXT M: NEXT N
399 REM ***  RESULTADO ***
400 K=0: IF D(LA+LB)=0 THEN K=1
410 FOR N=LA+LB-K TO 1 STEP -1
420 PRINT RIGHT$(STR$(D(N)),1);
430 IF FNM3(N)=1 AND N>1 THEN PRINT".";
440 NEXT
450 END
```







```
9 REM *** NUMEROS ***
10 INPUT "DIVIDENDO";A$
20 INPUT "DIVISOR";B$
99 REM *** INICIALIZACION ***
100 LA=LEN(A$): LB=LEN(B$)
110 DIM A(LA+1),B(LB),D(LA),Q(LA-LB+1),T(LB)
120 DEF FNM3(X)=ABS(X)-3*INT(ABS(X)/3)
199 REM *** MATRICES ***
200 FOR N=1 TO LA
210 A(N)=VAL(MID$(A$,LA-N+1,1))
220 NEXT
230 FOR N=1 TO LB
240 B(N)=VAL(MID$(B$,LB-N+1,1))
250 NEXT
299 REM *** DIVISION ***
300 FOR M=1 TO LA-LB+1: Q(M)=0: NEXT
310 FOR CA=LA-LB+1 TO 1 STEP -1
320 GOSUB 600
330 IF FL=-1 THEN 360
340 Q(CA)=Q(CA)+1
350 GOTO 320
360 NEXT
399 REM *** RESULTADO ***
400 K=0: IF Q(LA-LB+1)=0 THEN K=1
410 PRINT "COCIENTE DE ";A$;"/";B$;"=";
420 FOR N=LA-LB-K+1 TO 1 STEP -1
430 PRINT RIGHT$(STR$(Q(N)),1);
440 IF FNM3(N)=1 AND N>1 THEN PRINT ".";
450 NEXT
460 PRINT: PRINT "RESTO =";
470 K=0: IF A(LB)=0 THEN K=1
480 FOR N=LB-K TO 1 STEP -1
490 PRINT RIGHT$(STR$(A(N)),1);
500 IF FNM3(N)=1 AND N>1 THEN PRINT ".";
510 NEXT
520 END
599 REM *** SUBRUTINA DE RESTA ***
600 FL=0
610 FOR J=1 TO LB: T(J)=A(CA+J-1): NEXT
620 FOR K=1 TO LB
630 IF A(CA+K-1)>=B(K) THEN 660
640 A(CA+K-1)=A(CA+K-1)+10
650 A(CA+K)=A(CA+K)-1
660 D(K)=A(CA+K-1)-B(K)
670 NEXT K
680 IF A(CA+LB)>=0 THEN 720
690 FL=-1: A(CA+LB)=0
700 FOR J=1 TO LB: A(CA+J-1)=T(J): NEXT
710 GOTO 730
720 FOR K=1 TO LB: A(CA+K-1)=D(K): NEXT
730 RETURN
```

nes en un solo segundo, es realmente para ponerte los pelos de punta. Podemos pensar lo que supuso calcular ese 2 elevado a 251 que ejecutó el Cray-1 después de 32 horas, y meditar que si él tardó 32 horas, ¡qué hubiera tardado un hombre sin ordenador! Probablemente millones de años para hacer las cuentas. Sólo con este tipo de ordenadores podemos hacer cosas que jamás habríamos pensado y que nunca se hubieran hecho en la historia del mundo, muchas de ellas, como la seguridad de un avión, de trascendental importancia.

LOS FLOPS

Por curiosidad, vamos a calcular la velocidad exacta en megaflops de nuestro ordenador. Podemos hacer un programa que nos dé una idea de ella. En realidad, si nosotros lo hacemos en Basic, y ya que éste es interpretado, cada operación ha de ser «compilada» primero, y después ejecutada, el programa mediría realmente la velocidad del lenguaje, más que la del propio ordenador. Deberíamos hacerlo en ensamblador, que es el «lenguaje» más rápido que existe, aunque por supuesto el más engorroso y específico, pero ya que normalmente nuestros programas los hacemos en Basic es un lenguaje muy bueno para cálculos, ya que es muy simple, se pueden definir funciones con mucha facilidad y permite el paso de cadenas a números y viceversa con funciones predefinidas, pero tiene el grave inconveniente de su lentitud. De ahí que lenguajes como el FORTRAN sean los que realmente se usan a la hora de escribir un programa científico.

```
10 A=TI
20 FOR L=1 TO 1000: C=N*N:
NETX
30 B=TI
40 TIEMPO=(B—A)/50
50 F=1000/TIEMPO
```

```
60 PRINT «FLOPS»:F
70 PRINT «MEGAFLOPS»:F/1E6
```

donde **TI** es el contador que al encenderse el ordenador se incrementa cada 1/50 segundos, una vez por cada ciclo de frecuencia de la corriente eléctrica. La variable **A** almacena el valor del contador antes de efectuar los cálculos, y **B** en el instante en que ya se han analizado las operaciones. Su diferencia será el número de ciclos de corriente que hayan pasado, que al dividirlo entre 50 nos dará el número de segundos. (Recordemos que la corriente es de 50 ciclos por segundo.)

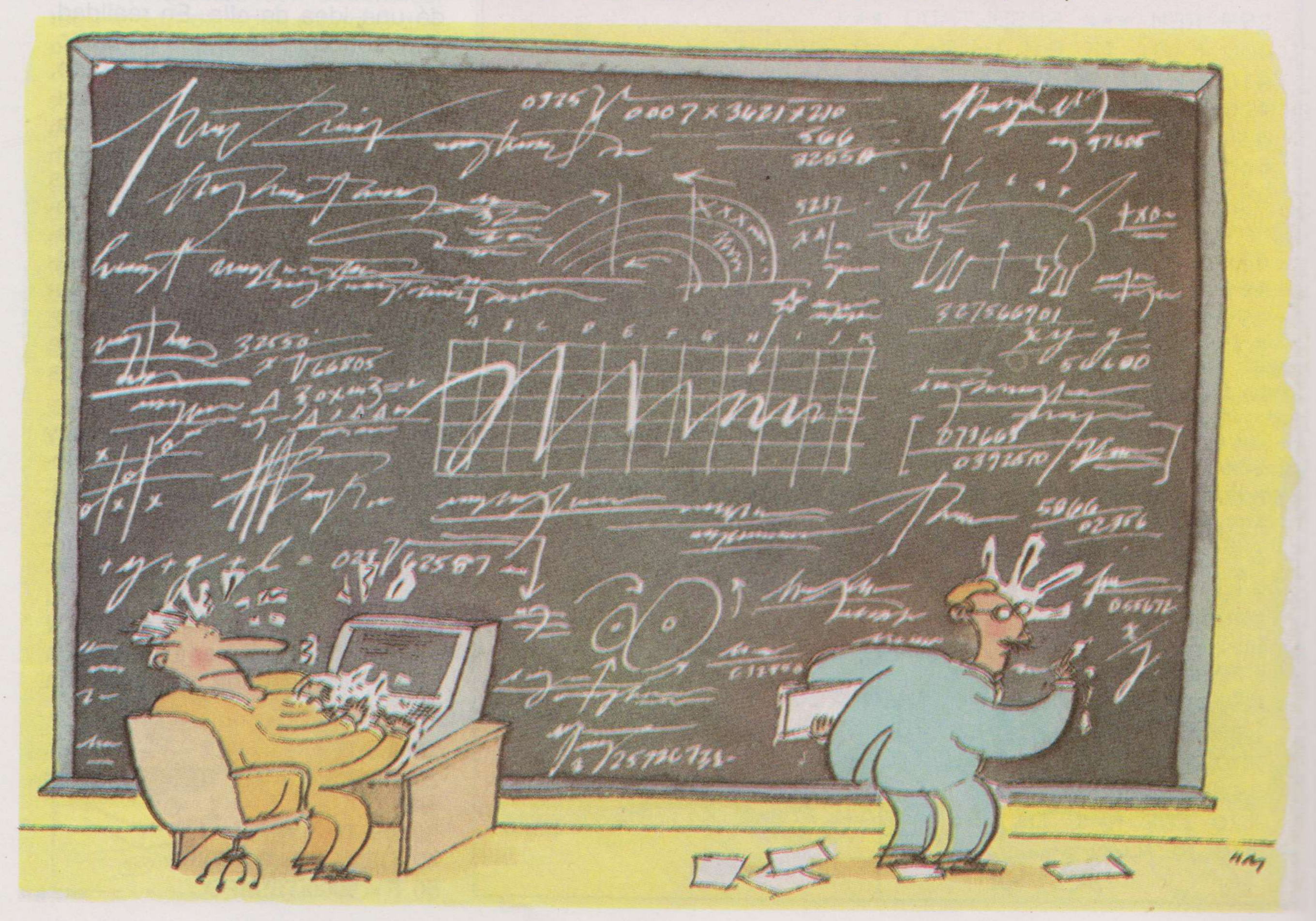
Si para hacer 1000 operaciones se ha empleado **F** segundos, en uno hará 1000/F operaciones, y será ese el resultado. Dividiendo entre un millón encontraremos los megaflops famosos.

Aproximadamente 184 flops, es decir, 0.00020 megaflops será la velocidad. Es muy poco, comparado con el Cray-1, pero podemos consolarnos pensando que si nuestro vecino se compra ese superordenador, tendrá que pagar más de tres mil millones de pesetas por tenerlo, y una factura de electricidad que le va a hacer quedarse de un cierto color amarillento.

MIS ENEMIGOS LOS DECIMALES

¿Cuál es el resultado de 11/17? Aproximadamente 0.65 sin pensarlo demasiado. Esta respuesta la mayoría de las veces es suficiente, pero en otras se requiere una precisión mucho más alta. Tal es el caso en el diseño de un aparato o la confirmación de una fórmula. Una simple calculadora no nos vale, ni siquiera un ordenador, haría falta que nos pusiéramos con un lápiz y un papel y lo hiciéramos nosotros mismos. En este caso nosotros dividiríamos el primer número entre el segundo (11 entre 17), y obtendríamos una cifra (el cero), y pondríamos debajo del dividendo el resto (que sería 11), solemos decir entonces que «bajamos un cero» al resto (con lo que queda 110), que no es otra cosa que multiplicar por 10, y dividimos otra vez entre el divisor (el 17). De nuevo sacaríamos otra cifra y el proceso se repetiría indefinidamente, pudiendo sacar así todas las cifras que quisiéramos. Esto es lo que exactamente hace el programa:

```
10 INPUT "X";X
20 INPUT "Y";Y
```



```
30 C = INT (X/Y)
40 PRINT X;"/";Y;" = ";
50 PRINT C;"/";
60 X = 10*(X-Y*C)
70 C = INT (X/Y)
80 PRINT MID$(STR$(C),2);
90 GOTO 60
READY.
```



Lo único resaltable es que la función MID que nos coge el segundo carácter del resultado, cada vez que dividimos el resto entre el divisor para evitar el primer carácter que corresponde al signo (blanco si es positivo).

Esto nos vale para cualquier par de números, siempre que sean de menos de diez cifras, y podemos hacer con igual facilidad una división que no sea de enteros, como 1,000.001 entre 17,0000017, ya que lo único que deberíamos hacer es prescindir de las comas (en este caso, el resultado de 1,000.001/17,0000017 es el mismo que el de 10.000.010/17.000.017), multiplicando por la potencia adecuada de diez. Podríamos sacar así todos los decimales que queramos, tantos como aparezcan hasta presionar la tecla RUN/STOP. Pero os daréis cuenta en seguida de que siempre existe un momento en que los decimales vuelven a repetirse, así, 2/110 es 0.0181818 repetido hasta el infinito. Esto no es nada nuevo, pero lo que si os puede resultar nuevo es que el número de decimales diferentes hasta empezar a repetirse, en una división de X/Y, es como máximo de Y-1, es decir, antes de y decimales.

MULTIPLICACIONES GIGANTES

Al principio del artículo veíamos cómo el Cray operaba con números de 69 dígitos. Esto para nosotros todavía no es posible, pues nuestro Commodore sólo nos permite actuar como mucho con números de nueve cifras. Eso mismo nos pasaba con el programa anterior, que era muy preciso pero sólo para números que cumplían esa condición, ya que sino, el ordenador redondea, introduce la notación exponencial y se pierden dígitos. Precisamente uno de los problemas intrínsecos de la informática es el del redondeo, y no sólo es antiguo, sino que además un problema que nadie podrá resolver jamás. Veamos por qué.

Los números que usamos son muchos. En realidad, entre dos números cualquiera como el 1 y el 1000 hay infinitos números incluso, entre 2,5 y 2,6 está el 2,55 y el 2,555, o el 2,5555, etc... Esto significa que el ordenador trabaja con un número restringido, un subconjunto de los números verdaderos («Reaies», como se les denomina en matemáticas).

Normalmente, para usos corrientes, nos basta almacenar los números en 16 *bits*. De esa forma podríamos guardar el número 0,00015 y el 0,000016, pero no los intermedios, por ejemplo. Si necesitamos más precisión, podemos usar números de cuatro *bytes*. Para guardar más decimales, pero siempre existe un límite. Suponeos que en el caso anterior dividimos 0,000015 entre 10, como el resultado es 0,00000015 no cabría redondearía a ceros todos los *bits*.

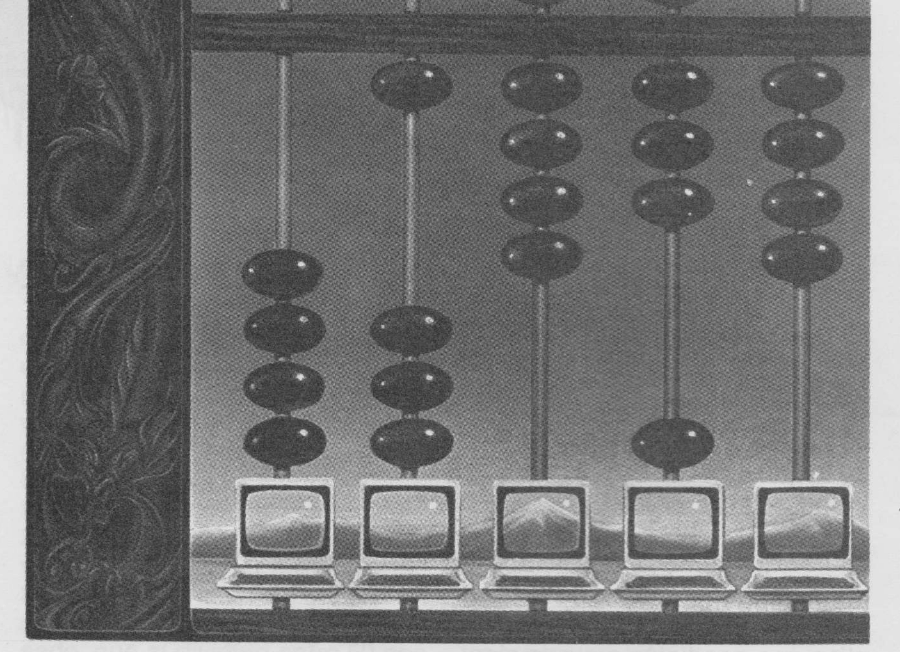
Este error de ordenador sería un error típico que podría dar lugar a otra serie de errores que, aunque estuviera bien programado, destrozarían los resultados.

Esto nos da idea de lo importante que es trabajar con muchas cifras en determinados casos, y por eso, vamos a intentar hacer dos programas que permitan multiplicar y dividir dos números sean cuales sean sus longitudes.

Está claro que cuando el programa nos pida los números, éstos han de entrar como una cadena, ya que si no, instantáneamente en la entrada del ordenador redondearía el número con lo que todo estaría perdido. Por eso los introducimos en *A$* y *B$*.

Una vez en *A$* y *B$ necesitamos coger número a número, y para eso definimos dos arrays, uno para A* y otro para *B* y cada elemento de ellos contendrá una cifra de *A$* y B$. Esta parte la lleva a cabo la subrutina 200, después de haber definido en la 100 las dimensiones con la única particularidad de que empieza a tomar desde el final cada uno de los dígitos, y así *A(1)* corresponde al último carácter de *A$ A(2)* al penúltimo, etc... La verdadera subrutina de multiplicación empieza en la línea 300, y el algoritmo que emplea es exactamente el mismo que el que empleamos nosotros cuando lo hacemos con un lápiz y un papel. Colocamos arriba, en el papel, uno de los números, por ejemplo, *A()* y en la parte de abajo del otro *B()*. Después la cifra menos significativa de B. esto es *B(1),* la multiplicamos por todas las cifas de arriba, es decir, de *A()* El resultado lo almacenamos en *D,* previamente inicializado a O.

Con la segunda hacemos lo



mismo, multiplicamos por todas las de *A* y vamos poniendo el resultado debajo de la fila anterior, pero desplazado un lugar, y sumamos ambas. La variable *MN* guarda el índice donde vamos a sumar el resultado, es decir, la cifra de *D()*, por ejemplo, la cifra menos significativa será de multiplicar *A(1) por B(1)* y estará colocado en *D(1 + 1-1) = D(1)*.

De esa manera, *D* se va incrementando con los productos de cada cifra de *B()* con todas las de *A()*, y en cada una de las veces, desplazamos un lugar. Podríamos resumirlo como: «multiplico una cifra de *A()* por *B()*, *desplazo y sumo a D()*, y prosigo así con cada cifra de *A()»*.

¿Y qué son esas *FNU(X)*, *FNC(X)* y *FNM3(X)?*

Son los que a veces se denomina «funciones módulo». Para dos números *X* y *N*, decimos *X* módulo *N* al número de veces que al resto de dividir *X* entre *N.FNU*, por ejemplo, es la función *X* módulo 10, y se usa ya que cuando hacemos una suma, es posible que nos «llevamos una», o sea que la suma total pase de diez, en cuyo caso ponemos la última cifra (el resto de dividir *X* entre 10) y pasamos una, dos... o para ser exactos, la parte entera de *X/10*, que es lo que deduce *FNC (X)*, a la siguiente columna.

FNM3(X) simplemente pone los puntos cada tres cifras, para que en lugar de salir 10098213 salga 10.098.213, y sea de más fácil comprensión.

DIVISIONES GIGANTES

De igual manera que antes, haremos el programa para dividir basándonos en un «algoritmo natural», que no es el que usamos nosotros pero que en el fondo se basa en lo que es la división.

Dividir un millón de pesetas entre 13 personas no es ni más ni menos que repartir un número equitativo de pesetas a cada una de las personas, y ver a cuánto toca cada uno.

Si no supiéramos dividir, lo que haríamos sería dar una peseta al primero, otra al segundo, otra al tercero, etc... hasta que los trece tuvieran cada uno una peseta. Nos quedan entonces un millón menos trece pesetas, que tendremos que seguir repartiendo. En cada caso restamos trece monedas, e incrementamos en uno lo que le toca a cada uno, hasta que el número total de monedas sea inferior a trece, en cuyo caso podremos decir que ese número es el resto, y que lo que tiene cada uno es el resultado. Eso es exactamente lo que hace el programa para dividir dos números, *A()* y *B()*, que hemos tecleado, igual que en el programa de multiplicación, en *A$* y *B$*. La subrutina de división, como antes, está también situada en la dirección 300, y su función es ir restando ambos números hasta que se cumpla que lo que queda es menor que el divisor. *FL* es un *flag* que devuelve la subrutina de resta, situada en la 600 y que está a -1 sólo cuando se llegue al final, en el que entonces en *A()* quedará el resto.

Lo único destacable de la subrutina de resta, que devuelve en *A()* el resultado de *A() - B()*, previamente asignado a una matriz *D()*, es que ha de tener en cuenta si la cifra a restar, *A(I)*, es menor que *B(I), en cuyo caso actúa como lo hacemos nosotros, es decir, 2-7, lo traducimos a «de 7 a 12 van 5 y me llevo 1», y es esa una la que resta en la columna siguiente (A(I + 1))*.

EL RETO

Ya está. Ya somos capaces de multiplicar los números más grandes y de dividir las cifras más tenebrosas. El mundo de los grandes números está a nuestro alcance, y vamos a empezar con algo fácil. Por ejemplo, vamos a comprobar si 2^{251}-1 es primo o no, igual que lo hizo el hermano mayor, el Cray 1. No obstante, no vamos a emplear la misma forma, sino que vamos a aprovecharnos de que el Cray descubrió los tres factores que daban ese número, que son:

178230287214063289511
61676882198695257501367
12070396178249893039969681

Y los multiplicaremos. Podéis comprobar que el resultado es 2^{251}-1 con la subrutina de multiplicación más un pequeño programa para hacerlo. Os llevará algo de tiempo, pero mucho menos que el que le llevó al Cray, y demostraréis también que Mervin Mersenne estaba equivocado. Pero

¿Sois capaces de sacar un primo mayor?

Os retamos desde aquí a que hagáis un programa que calcule si un número es primo o no, y en este último caso, cuáles son los factores.

Nota.- El programa son seis líneas, aparte de la subrutina de multiplicación, pero no es tan obvio como a primera vista parece.

ANUNCIESE por MODULOS

LOBERCIO
COMPUTER - CENTER

Unico en España:

- DISPONEMOS TODO TIPO DE REPUESTOS Y MANUALES DE REPARACION DE COMMODORE
- REPARACION RAPIDA Y GARANTIZADA
- TODA CLASE DE PERIFERICOS 64/128
- TONER PARA COPIADORA CANON- MINOLTA

Consultas: Tel.: (952) 33 27 26
Avda. de Andalucia, 17 29002 MALAGA

MADRID
(91) 733 96 62
BARCELONA
(93) 301 47 00

La industria informática española tiene lo que necesita.



Publinformática, S.A. • Bravo Murillo, 377, 3º E • Telf. 733 96 62/96 • Madrid 28020

Las palancas de juego han sufrido incontables mejoras desde sus principios. Antes, respondían por simples contactos sin ningún tipo de interruptor que desactivase esa acción. La vida de estos aparatos no dependía de los contactos, sino del material de sus palancas. Fue con los primeros interruptores cuando empezaron a recortarse la vida de estas palancas de juego, llamadas *joysticks.*

Entre los primeros interruptores, conocimos unas pequeñas chapas de acero, de espesor muy fino, circulares y cóncavas que situadas sobre un circulo integrado, que al ser presionadas en su parte superior hacían de interruptores.

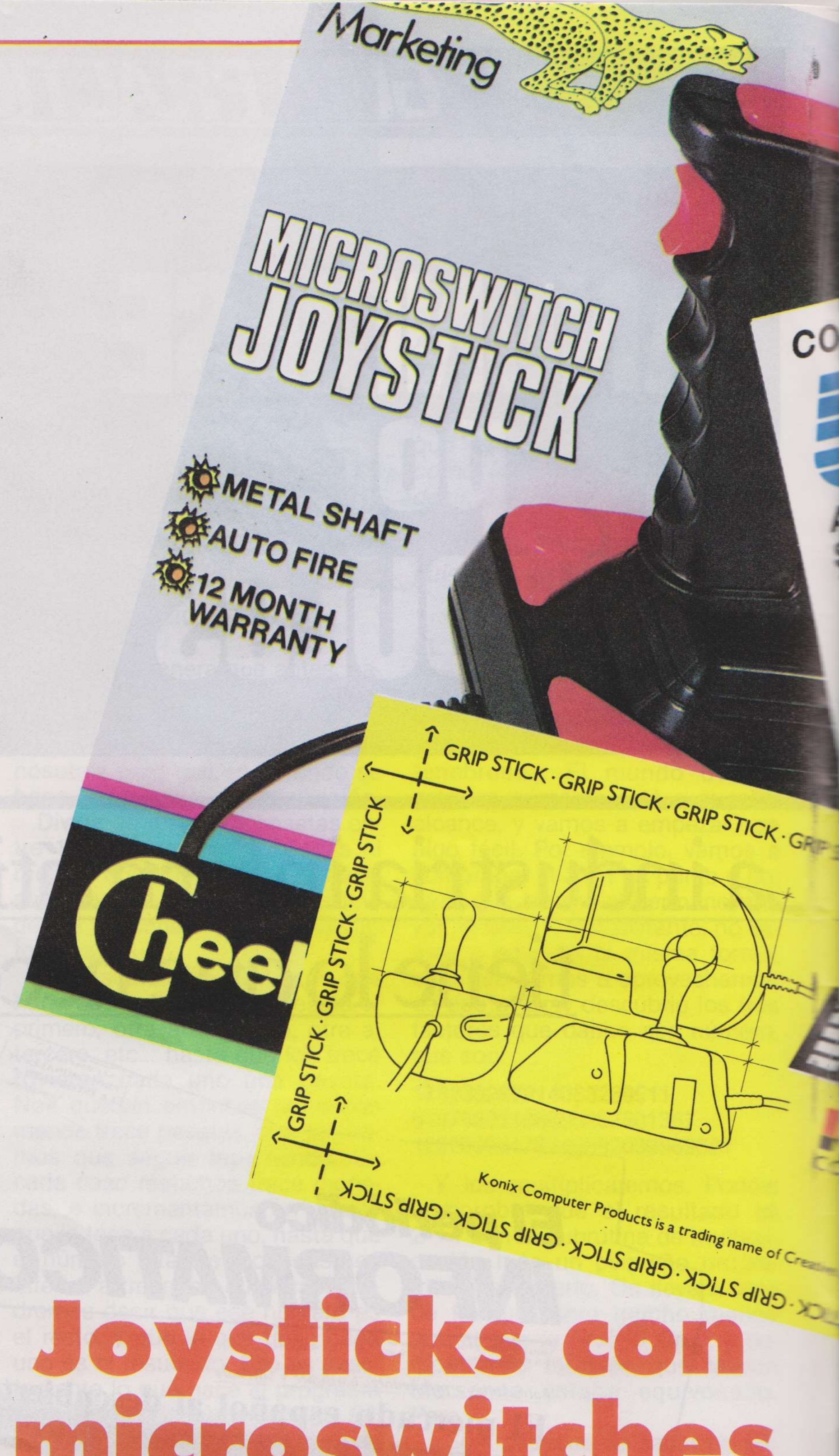
Estas especies de lentejas eran generalmente, la primera parte del *joystick* que comenzaba a fallar.

Los que tuvísteis algún *joystick* hace un par de años, seguro que vísteis en su interior estas pequeñas chapas, y también casi seguro, que si le dísteis un trato duro, a los tres meses ya observaríais irregularidades en su funcionamiento.

La vida de este tipo de interruptores suele ser muy corta, las continuas presiones hacen que la chapa termine abriéndose y por lo tanto descontrolar la señal.

Casi paralelo a este tipo de interruptores, evolucionaron otro tipo de *joystick*, con una clase distinta de interruptor, al que llamábamos de palanca. Este consistía en una lámina de acero de forma rectangular, que estaba sujeta por un extremo y era presionada por el otro cuando se realizaba el contacto.

La duración de estos interruptores era mucho mayor, pero entre los usuarios, estos *joysticks* eran considerados como lentos y faltos de tacto para saber cuándo se emitía la señal.



Joysticks con microswitches

La experiencia, una vez más, obliga reforzarse a nuestros inagotables compañeros de juego. Difíciles luchas, en las que debe de participar hasta el final, sugieren un material resistente y últimamente hasta hacen necesaria la incorporación de microswitches.



La incorporación de *microswitches* en los *joysticks*, alarga considerablemente la vida de éstos, pues el desgaste es menor y en principio las posibilidades de fallo se reducen. De todas formas si alguno de estos *microswitches* se estropea, también cabe la posibilidad de quitar un tornillo, buscarlo en una tienda especializada y cambiarlo por otro.

La calidad de los *microswitches* varía según el tipo y el material de fabricación, podemos encontrarlos de acero con muelle, o simplemente de plástico. También es importante la forma que tiene el palo o *stick* de golpear estos *microswitches* y los topes que centralizan el desplazamiento.

Otra de las novedades que podemos apreciar en este tipo de *joysticks* es la carencia de circuitos integrados, es decir, la clásica tarjeta verde que se encontraba dentro de la carcasa de plástico del *joystick*. Ahora, los *microswitches* se sujetan a la propia base o a la parte superior de la carcasa, evitando piezas innecesarias.

La mayoría de los *microswitches* está cubiertos por una carcasa que les protege del polvo y de los residuos que deja el desgaste del plástico en los continuos roces con la rótula.

Si abriéramos uno de estos *microswitches* la sorpresa que nos llevaríamos sería grande, pues la simpleza de su interior no responde a lo que, en apariencia se puede ver, dos pequeñas chapas una de ellas curvada, hacen de interruptores.

Toma especial importancia un baño de oro que debe de tener en los contactos.

Con esto simplemente hemos querido mostrar lo que puede llamarse nueva generación de *joystick* con *microswitches* en cualquiera de sus versiones, con autofuego, ergonómicos, etc.

CHEETAM MATCH I

Compatible Sinclair ZX Spectrum, Commodore 64/128, VIC-20, MSX, Atari y Amstrad. Incluye la novedad de un doble conector que le hace adaptable al nuevo Sinclair Spectrum Plus 2.

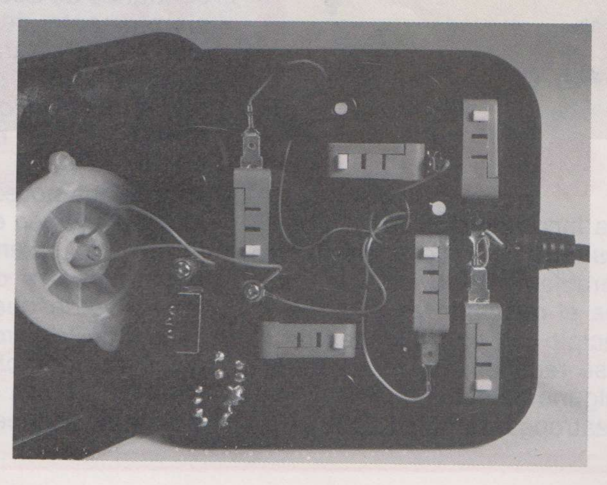
El diseño de este *joystick* responde a los clásicos ergonómicos de puño, incluyendo la novedad de cuatro botones de disparo, dos en la base, para jugar a dos manos y dos en el puño para jugar con una mano. La base tiene cuatro ventosas que sirven para adherir el *joystick* a una mesa, método habitual entre los ya conocidos. El mando incluye una base metálica que impide que por alguna caída o mal uso, éste pueda romperse. Incorpora cuatro ventosas en la base y un



autofire gobernado por un *chip*.

Los seis *microswitches* que lleva están protegidos por unas carcasas verdes de plástico y están sujetos a la base del *joystick*, careciendo de la clásica placa de circuito integrado. Una de las pegas que encontramos fue que el botón de *autofire* es desmontable y que los contactos de los dos botones de disparo del mango son de palanca. Por lo demás responde muy aceptablemente, aunque la rótula y los contactos con los *microswitches* son de plástico.

Por un precio de 4.000 ptas. aproximadamente puedes solicitar este robusto *joystick* en British Soft. Tel. (91) 246 81 53, o adquirirlo en alguna tienda especializada.





Compatible con Atari, Amstrad, VIC-20, Commodore 64/128, SVI-318/328, Sinclair ZX Spectrum.

El Quick Shot II fue uno de los *joystick* más vendidos en los últimos años, apareciendo ahora como Quick Shot II Plus, con la novedad de *microswitches* y un preciso *autofire*.

Como casi todos sabemos, posee un mando ergonámico adaptable a la mano y las populares ventosas que hacen de él un *joystick* manejable con una sola mano.

Son seis los *microswitches* que incorpora, cuatro para las direcciones o caminos y dos para los botones de disparo situados en la empuñadura.

Interiormente posee la clásica placa con el circuito integrado, a la que están adheridos los cuatro *microswitches*, los cuales, a diferencia de los demás no están cubiertos por la típica carcasa. Estos también se diferencian en el mecanismo de contacto, un muelle hace las veces de interruptor, desplazando una placa metálica. De aspecto muy duradero y evitando los plásticos desgastables se prevé una larga vida a este *joystick*.

Posee un *autofire* controlado por un *chip*, método de gran utilidad para determinados juegos. Lo único criticable puede ser la rótula de plástico, base de la empuñadura que como pudimos ver en los modelos anteriores es fácilmente vulnerable a los malos tratos.

Es uno de los grandes clásicos y por un precio de 2.500 ptas. aproximadamente puedes adquirirlo en Hispasoft, S.A. Tel. (976) 39 99 61 o en cualquiera de sus distribuidores.

COMPETITION PRO 5.000

Compatible con Amstrad, MSX, Atari, Commodore 64/128, VIC-20 y Sinclair ZX Spectrum, podemos considerarle como uno de los *joysticks* más duros del mercado.

Con un mando de bola, bastante distinto a los habituales, y un nuevo diseño que responde a un uso menos agresivo. No posee las clásicas ventosas en su parte inferior, encontrándose en su lugar unos tacos de goma, lo suficientemente adherentes, por las fuerzas que sobre este *joystick* se desarrollan.

En su interior, podemos observar cuatro *microswitches* protegidos por una caja de plástico azul que por un tornillo se sujetan a la parte superior de la carcasa exterior del *joystick*, encargándose éstos de transmitir las ocho direcciones posibles.

Además encontramos incorporados dos contactos de doble palanca, que hacen las funciones de interruptores para los botones de disparo.

En conjunto podemos decir que es un buen *joystick*, del que hay que resaltar su dureza ante los malos tratos y su mando de bola que obliga a utilizar las dos manos en el juego; una para disparar y sujetar y la otra para marcar la dirección.

A un precio de 3.900 ptas. aproximadamente puedes adquirirlo en Hispasoft, S.A. Tel. (976) 39 99 61 o en cualquier tienda de informática.



SPEEDKING

Compatible Sinclair ZX Spectrum, Commodore 64/128, Atari, Dragon, MSX y Amstrad. Es un *joystick* que renueva totalmente el concepto de posición física en el juego, pues no se apoya sobre una mesa, si no que se sujeta con la mano izquierda, y apoyado preferentemente en el cuerpo se juega con la mano derecha. El botón de disparo se encuentra en la parte inferior del *joystick* y está diseñado para pulsarlo con el dedo índice de la mano izquierda.

Los interruptores que posee son cinco *microswitches,* dispuestos cuatro de ellos para las direcciones y el quinto para el disparo. El mango está preparado para agarrarse con tres dedos, lo que evita desahogos violentos contra el *joystick*. El interior del mango es de hierro y da un aspecto muy resistente.

En la caja podemos leer una aclaración que especifica, que este *joystick* también es fabricado para ser utilizado por personas zurdas, es decir, con el botón de disparo a la parte izquierda del *joystick.*

Si estás dispuesto a adquirir este *joystick* ergonómico, puedes hacerlo en Serma. Tel. (91) 256 21 01 o en cualquier establecimiento distribuidor.



¿te interesa?

Vendo C-64, *cassette*, impresora SEIKOSHA GP-500 VC, TV 14' B/N, cintas comerciales, revistas con cientos de programas, libros praa C-64, *joystick* y utilidades todo por 82.000 ptas. Dirigirse a: José Pascual. Tel.: (93) 896 09 42. Sant Pere de Ribes.

¿

Deseo contactar con usuarios del CBM-64 para intercambio de *software* (poseo 500 títulos) en disco o cinta así como conocimientos y material de código máquina. *Diskette* preferentemente. Mandar lista a: Miguel Pérez Méndez. C/ Amboto, 5, 2 Izq. Algorta. Vizcaya.

¿

Vendo cintas originales de juegos novedosos en perfecto estado a 1.000 ptas. cada una. Son: Hardball. Friday 13. Chostbusters. Decathlon. Exploding Fist. Rambo. Saboteur. Uridium. On Court Tennis. México 86. Hulk. Knight Rider. etc. Llamar a Luis Miguel Montero Gómez. Tel.: (91) 471 14 28. C/ Alejandro Sánchez. 8. 28019 Madrid. Todos originales.

¿

Vendo Vic-20 en perfecto estado + ampliación de 3K, Superexpander + ampliación de 16K + *cassette* para el ordenador + *joystick* + 5 cintas de juego y dos cartuchos de juegos. Además regalo varias cintas con programas y juegos hechos por mí y un lote de 17 revistas «Commodore Magazine» + dos libros de introducción al Basic en el Vic-20. Todo por 25.000 ptas. Dirigirse a: Andrés Fco. Giménez Sendra. C/ Padre Urbano, 20-24. 46009 Valencia. Tei.: (96) 365 43 63.

¿

Oferta especial. Vendo 128 nuevo un mes de uso, manuales en castellano, disco CP/M, juegos en cinta Hardball, Starquake, Indiana Jones y numerosos juegos escritos, más Vic 20, *datassette* cartucho memoria 16K, ajedrez, juegos en cinta, comecocos, Heitik, Topo, Tron, Escalador, Desastre, etc., y numerosos programas escritos. Un libro de Basic y otro de programación por sólo 60.000 ptas. Llamar de 17,15 a 20,15 a Juan Carlos González. Tel.: (942) 21 72 00. C/ Canalejas, 47. Santander.

Desearía intercambiar programas de juegos y utilidades, para el Commodore 64 en *cassette*. Enviar vuestra lista. Contestaré a todos con la mía. Interesados dirigirse a: Isidoro Montero Calzado. C/ San Ignacio, 10, 3 Dcha. Alcorcón. 28921 Madrid.

¿

Unos amigos hemos formado un Club para el intercambio de *Software* del C-64 y del C-128. Pensamos cambiar 3*1. sólo usuarios de Zaragoza. Escribid a: David Aso Palacín. Residencial Paraíso, 2, E-D, 4.°-C. 50008 Zaragoza. O bien llamar al (976) 21 95 64 (preguntar por David. Tardes o al (976) 45 32 49 (Eusebio. Mañanas).

¿

Para los poseedores de un Vic-20, vendo un *interface* de CW y RTTY de la marca VIC-HAM con sus programas originales en código máquina. El precio de venta es de unas 13.000 ptas. Interesados dirigirse a: Ramón Petit. Feixa Llarga s/n. L'Hospitalet. 08907 Barcelona. Tel.: (93) 336 26 39.

¿

Vendo Commodore Plus/4. 64K RAM + *Datassette* + *Joystick* + 10 juegos + 2 Manuales (en inglés); todo en perfecto estado, por 20.000 ptas. Interesados llamar a: Alejandro Alvarez. Tel.: (93) 350 44 74 de Barcelona (horario de comidas).

¿

Vendo Commodore 64, *cassette*, revistas, libros: «Qué es, para qué sirve, y cómo se usa»; «Gráficos y Sonidos» (con cinta). *Joystick* y cinta de juegos por 40.000 ptas. Llamar a: Javier Fernández a partir de las 21 horas. Tel.: (93) 359 83 93. Barcelona.

¿

Vendo Commodore 128 (Agosto 1986) a 40.000 ptas., 1541 (30.000 ptas.), Riteman C+ (35.000) o junto por 90.000 ptas. Regalo programas profesionales. Tel.: (93) 725 45 34, preguntar por J. López.

Cambio programas en cinta para CBM-64. Escribir a: Joan Rimbas. C/ Lepant, 5, 5-1. 43202 Reus.

¿

Intercambiamos programas en cinta o disco. Club Commodore Canario. Apartado 2485. 35080 Las Palmas de Gran Canaria.

¿

Cambio todo tipo de programas tanto para el C-64 como para el C-128. Tengo una buena cantidad de programas para el intercambio. Mis datos son: Antonio Dorta. Apartado de Correos, 184. La Cuesta. La Laguna. Tenerife.

¿

Interesa comprar impresora MPS 801, Seikosha 6P 100 VC o semejante. Marti, Vincenc Algueró Ruiz. Ctra. de Mata, 54, 4, 2. 08304 Mataró (Barcelona). Tel.: (93) 796 21 70.

¿

Cambiamos programas para el C-64. Escribir al apartado de correos 291 de 43080 Reus o a Joan Rivas. C/ Lepant, 5, 5-1.ª. 43202 Reus.

¿

Vendo Commodore 64 completamente nuevo (garantía 4 meses) + *datassette* + *joystick* + 25 juegos + curso de informática. Precio a convenir. Joan Ricard Solá i Godoy. C/ Mare de Deu de Port 375, At.°-2.ª. 08004 Barcelona. Tel.: (93) 331 83 76.

¿

Vendo ordenador Commodore 64, *datassette* C2N y 4 programas (Simon's Basic, Commando, Summer Games, Drop Zone), en buen estado (1 año), con las cajas de embalaje e instrucciones por sólo 38.000 ptas.; con 80 programas 45.000 ptas. Interesados llamar a: (93) 674 68 64 o escribir a Albert Galvany Matas. P. Creu, 33, 3.°-3.ª. San Cugat del Vallés (Barcelona).

Deseo intercambiar programas de juegos y utilidades para CBM-64 en cinta. Escribir a: Carlos Bardera. Pza. del Horno, 6. Palamós. 17230 Gerona, o llamar al (972) 31 44 27.

Atención. Cambio o vendo programas para Commodore 64, 128 y Amiga. Sólo en disco. Interesados enviad lista a: José Luis López Costa. C/ Abasota, 16, 2.°. Algorta. Vizcaya. Tel. (94) 469 06 25.



Vendo colecciones de las revistas: Commodore Magazine (n.° 1 al 33) y Commodore World (n.° 1 al 13). guía de referencia del C-64 y Manual de Cassette para Commodore 64 y Vic 20. Regalo programa de Ajedrez Colossus Chess 4.0. Llamar al (976) 23 31 13 de Zaragoza y preguntar por Javier.

Intercambio juegos y utilidades para el Commodore 64, en cintas o discos (preferentemente en discos). Prometo contestar, enviad lista a: Antonio Molina García. Barrio de la Constitución, B1. 12, 2.°-C. Melilla.



Vendo Commodore Vic-20 (1985), transformador y cables. Perfecto estado. Sin usar. Precio a convenir (barato). Interesados contactar con: Raúl Alegre Latorre. Avda. Valencia, 6. Utrillas (Teruel). Tel.: (974) 75 70 29.



Por cambio de ordenador vendo C-64 por 40.000 y una impresora MPS 801 por 26.000. Simon's Basic Card. + HesMon (Ensamblador) Card. Cada uno a 4.000 ptas. Cobol C-64 & C-128 sin necesidad de CP/M a 15.000 ptas. (disco y manual). Envío contra reembolso. Dirigirse a: Antonio Moreno Cintado. Apto. 315. Mérida (Badajoz). Tel.: 31 60 52.

Casi, casi nuevo vendo C-128 por 60.000 ptas. con todo completo. Dispongo además de más de 1.000 programas para cambiar con el comprador. Dirigirse a: Miguel Ortega Morales. Les Fogaroses, 64. Matadepera. 08230 Barcelona. Tel.: (93) 787 15 22.



Urgente, se cambia por unidad de disco 1541 ó 1570, un *datassette* para Commodore (PVP 10.000) más *interface* para copiado de programas con dos *datassettes* (PVP 5.000), más cartucho Soccer (PVP 6.000), más reset, especial para desbloquear programas, con programa especial para su funcionamiento (PVP 1.000), más Simon's Basic con instrucciones (PVP 14.000), más 200 programas entre utilidades y juegos (PVP 150.000). Daría también la pequeña diferencia en dinero. Mi dirección es: Antonio José Miranda. Apto. Correos, 71. C/ Ferrán de Querol, 24. 43830 Torredembarra. Tarragona.

ANUNCIOS GRATUITOS

Todos los anuncios (compras, ventas, cambios o comunicaciones de clubs de usuarios, etc.) que van en esta sección, deben tener un máximo de cuarenta palabras. Con el fin de facilitar la transcripción de los anuncios hemos recuadrado cuarenta espacios para que en cada uno vaya una palabra. Después, recortar y mandar a:

commodore Magazine
ANUNCIOS GRATUITOS
C/ Bravo Murillo, 377, 5.° A
28020 MADRID



DELETE

Esta corta rutina te permitirá borrar rápidamente y de una sola vez varias líneas de un programa BASIC.

Su utilización es muy sencilla. En la línea 60000 debes colocar el número de la primera línea que quieras borrar menos el incremento, en la línea 60001 debes colocar el número correspondiente al incremento que siguen las líneas de tu programa y en la línea 60002 el número de la última línea a borrar más el incremento.

Para sacar realmente provecho de este truco es aconsejable utilizarlo, como parte de un programa, situado al final de todo tu listado y seguido de una línea con END.

```
10 REM
20 REM
30 REM                        LINEAS A BORRAR
40 REM
50 REM
59999 END
60000 I1=0:REM ****** LINEA INICIAL - INCREMENTO *****
60001 I1=I1+10:REM *********** INCREMENTO ************
60002 IF I1=60 THEN END:REM LINEA FINAL+INCREMENTO ***
60003 PRINT CHR$(147):PRINT:PRINT
60004 PRINT I1:PRINT"I1="I1":GOTO 60001":PRINT CHR$(19)
60005 FOR I2=631 TO 533 :POKE I2,13: NEXT: POKE 198,3
```

Decelerador

Con este programa podrás relentizar todos los programas BASIC incluyendo LISTados. La rutina arranca haciendo un RUN que introduce por POKEs, los valores que hay en los DATAs. Para relentizar alguna acción, sólo tienes que escribir SYS 49152, y para volver a la velocidad normal, pulsar las teclas RUN/STOP y RESTORE.

Este programa añade una nueva rutina a la rutina de servicio de interrupciones, redireccionando los punteros de los vectores $314 y $315, decelerando el código y consiguiendo una pausa por un período de tiempo predeterminado, para luego retornar a la dirección $EA31. La rutina se ajusta a los tiempos X Y, pudiendo ser ajustado por algún valor entre 1 y 255. El valor X utiliza la dirección 49173, y el valor Y utiliza la 49178, para conseguir la pausa.

```
10 Y=0
20 FOR X=49152 TO 49183
30 READ A
40 Y=Y+A
50 POKE X,A
60 NEXT
70 IF Y<>4188 THEN PRINT"ERROR":STOP
100 DATA 120,162,015,160,192
110 DATA 142,020,003,140,021
120 DATA 003,088,076,174,167
130 DATA 162,000,160,000,200
140 DATA 192,010,208,251,232
150 DATA 224,255,208,244,076
160 DATA 049,234
```



LE OFRECE LOS MEJORES LIBROS PARA SU ORDENADOR



P.V.P. 750 PTAS.
(IVA INCLUIDO)
Descubre los misterios de la programación de una forma sencilla, con ejemplos, programas y organigramas.
(110 páginas, tamaño 13,5 x 21)



P.V.P. 800 PTAS.
(IVA INCLUIDO)
Con utilidades, juegos explosivos y gráficos dinámicos que lleva al BASIC hasta el mejor aprovechamiento de sus posibilidades.
(200 páginas, tamaño 15,5 x 21,5).



P.V.P. 750 PTAS.
(IVA INCLUIDO)
Un libro especialmente dedicado a los que se inician por vez primera en el mundo del Spectrum.
(100 páginas, tamaño 13,5 x 21).



P.V.P. 800 PTAS.
(IVA INCLUIDO)
Una inestimable ayuda que complementará la que proporciona el manual del ordenador.
(108 páginas tamaño 13,5 x 21,5).



P.V.P. 900 PTAS.
(IVA INCLUIDO)
Un compendio de los programas más diversos con los que podrá aprender jugando las importantnes características del BASIC.
(258 páginas, tamaño 15,5 x 21,5).



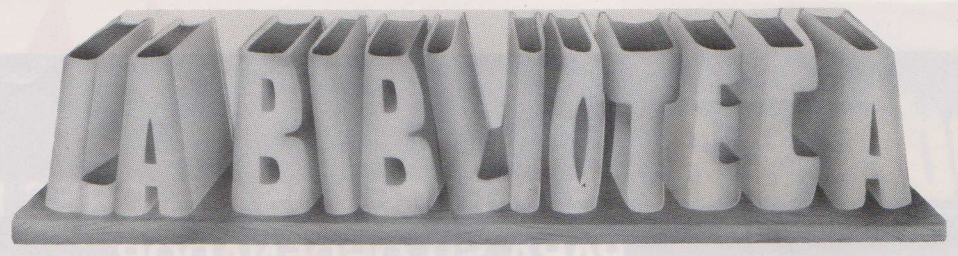
P.V.P. 800 PTAS.
(IVA INCLUIDO)
Muestra una visión más completa del correcto funcionamiento del juego de instrucciones del C-64.
(108 páginas, tamaño 13,5 x 21,5).

CUPON DE PEDIDO

enviar a:
infodis, s.a.
C/BRAVO MURILLO, 377
28020 MADRID

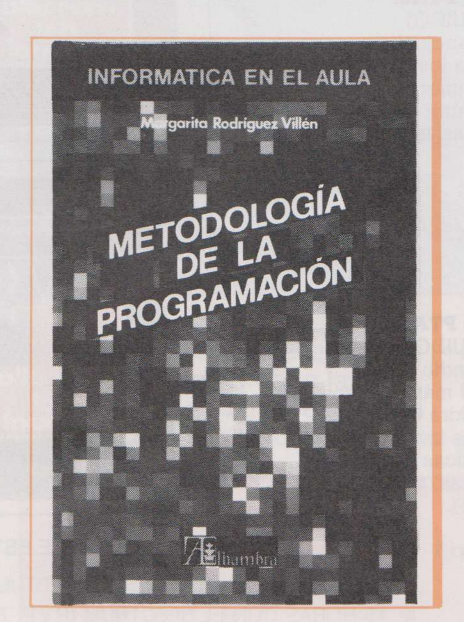
COPIE O RECORTE ESTE BOLETIN DE PEDIDO.

DESEO RECIBIR LOS SIGUIENTES TITULOS:
15 HORAS CON EL SPECTRUM (P.V.P. 750) ☐
LOS MEJORES PROGRAMAS PARA EL ZX SPECTRUM (P.V.P. 900) ☐
LOS MEJORES PROGRAMAS PARA EL COMMODORE 64 (P.V.P. 800) ☐
EL 64 MAS ALLA DEL MANUAL I (P.V.P. 800) ☐
EL 64 MAS ALLA DEL MANUAL II (P.V.P. 800) ☐
(más 100 ptas. de gastos de envío).
El importe lo abonaré POR CHEQUE ☐ CONTRA REEMBOLSO ☐ CON MI TARJETA DE CREDITO ☐ American Express ☐ Visa ☐ Interbank ☐
Número de mi tarjeta: ________________
NOMBRE ________________
CALLE ________________
CIUDAD ________________
PROVINCIA ________________ C. P. ________



Metodología de la programación.
Autora: Margarita Rodríguez Villén.
Editorial: Alhambra.
140 páginas.

Método. Inconscientemente pensamos en rigidez. Si la palabra es metodología nos sugiere algo monótono. Pero la programación, como cualquier otra actividad humana, requiere el seguimiento de una serie de pasos que nos lleven a un óptimo resultado. El libro de Rodríguez Villén se inicia con algo que parece tan obvio como es definir el concepto de información. Obvio pero que está en la propia base de la Informática (Información-automática). Tras una breve descripción de los tipos de programas con que nos podremos encontrar (lineales, cíclicos, alternativos...) se abordan las representaciones por antonomasia de la metodología de la programación: los diagramas de flujo, organigramas y ordinogramas y sus símbolos de representación. La información también la podemos representar de una forma tabular. Es una forma clara, concisa y fácil de leer y representan todas las variables de un problema. Los componentes y organización de una tabla de decisión son el objeto del tercer capítulo. Se describen sus tipos de entradas y clasificación, los tipos de reglas, la combinación de las tablas de decisión con ordinogramas y la forma de construir una tabla de decisión siguiendo los métodos progresivo y sistemático. Se cierra el capítulo con un análisis de tablas de decisión. Pero programar no conlleva únicamente la realización de un algoritmo, sino también la codificación en un determinado lenguaje de programación. Cada día se hace más necesario que las fases previas a la programación y la documentación a que da lugar sea lo más clara posible y que se realicen programas de fácil acceso a su información, en el momento en que sea preciso. La autora revisa los conceptos, elementos fundamentales y características de tres tipos de programación:



programación estándar, programación modular y programación estructurada. El siguiente capítulo aborda el tema de los ficheros, esto es, la estructura de la información. Los tipos de ficheros, las operaciones sobre los mismos y su estructura lógica se acompañan de abundantes ejemplos que facilitan la comprensión de los conceptos expuestos. El libro se cierra con un capítulo sobre análisis y programación en el que se describen las fases previas a la programación; el análisis y sus partes y la metodología utilizada.

El Commodore 64 como traductor. Traducción de idiomas extranjeros con el ordenador.
Autor: Heift.
Editorial: Data Becker.
Traducido del alemán. 273 páginas.

Al referirnos a los lenguajes y el ordenador, quedaba patente que hacíamos referencia a un lenguaje específico de programación que permitiera el «diálogo» entre el usuario y la computadora. Sin embargo, las ventajas y facultades del ordenador, su memoria, su rapidez, etc., pueden utilizarse para la traducción de textos. El libro de Data Becker no constituye un libro de consulta, sino un libro de texto en el que se detallan los pasos y procedimientos que nos conducirán a convertir el Commodore 64 en un magnífico traductor de idiomas. En los primeros capítulos se indica cómo debe la máquina distinguir entre los idiomas (en los ejemplos inglés y español), cómo «aprende» vocablos y la transformación de las palabras en números. La segunda parte del libro propone una elaboración de un sistema de programa léxico. Se detalla, en primer

lugar, el modo de organizar la grabación de vocablos o su respectiva traducción para que el tiempo de intervención sea corto. El autor opta por grabar los vocablos en una matriz bidimensional, grabando únicamente las traducciones, no la palabra a ser traducida. La palabra que ha de ser traducida sirve para conocer el índice en el que se deberá grabar la traducción en la matriz. De este modo se elabora un programa de grabación que permite introducir las traducciones de forma sencilla en el ordenador y grabarlas en disco. Seguidamente nos encontramos con un programa de lectura adaptable a ese programa de grabación. Ambas partes del



programa —programa de grabación y programa de lectura— quedan resumidas en un programa Menú, llamando a las distintas operaciones mediante la elección de unos códigos determinados. Tras este diccionario electrónico, el libro aborda la parte fundamental: traducir textos mediante el ordenador. La elaboración de un programa para la edición del texto fuente y del texto objeto lleva por nombre Textsal y forma parte del más ambicioso COMPTRANS (de Computeried Translation) que permite entrar texto fuente, traducir texto fuente, editar texto y afinar el texto en bruto. El libro concluye con una pequeña incursión en la traducción automática de artículos determinados-indeterminados, adjetivos, verbos auxiliares, objeto directo-indirecto... y con el deseo de que el propio lector investigue en el campo de la traducción automática.



cartas

Errores de impresión

P: Tengo un problema con uno de los programas editado en su revista Commodore Magazine, de diciembre n.º 34, PING-PONG, que por deficiencias de impresión no se aprecian con claridad las siguientes líneas del programa: 60860, 60870, 61010. Por deducción propia asigné unas datas a estas líneas y me da un error.

? ILLEGAL QUANTITY ERROR IN 50150.

Con la ilusión de que me puedan solventar este problema a la mayor brevedad posible, aprovecho la ocasión dándoles las gracias para saludarles.

Luis Granados
Barcelona

R: En la página 33 de la revista número 34 del mes de diciembre podemos apreciar tres pequeñas lagunas en la impresión. Pedimos disculpas por este tipo de errores y pasamos a listarles las líneas que no se entienden.

```
60860 DATA A9,00,85,FC,4C,B1,39,A9,01
60870 DATA 4E,37,A9,26,8D,41,36,A9,FF
60880 DATA 4E,37,C9,FF,D0,F9,A9,26,8D
60890 DATA 8D,2D,36,A5,FD,8D,4E,37,A9
```

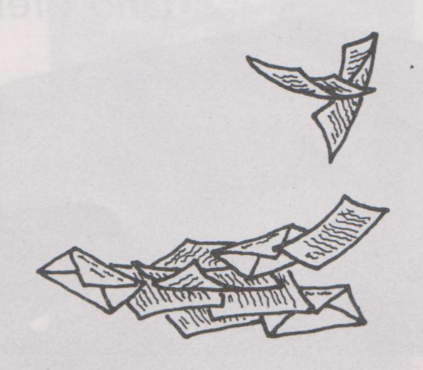
Y al final en la línea 61010 es posible que no se puedan apreciar los números 47,4F,20 situados al final de la misma. También en la línea 61020 los números 20,20,20,00,6A.

Terminal RTTY

P: Quisiera si es posible que me respondieran, bien a través de la revista o mediante carta, sobre la posibilidad de transmisión y recepción de RTTY (teletipos) con el Commodore-64, los programas que pueden utilizarse y si hay que hacer alguna conexión especial.

Casiano López
Murcia

R: La transmisión y recepción de teletipos con el Commodore-64/128 es posible a través de un interface «RTTY terminal» que distribuye Cimex Electrónica. El interface en código morse, capaz de decodificar RTTY, CW y ASCII desde 110 a 300 baudios y con un modo de seguimiento de la señal en CW y RTTY. Estas son parte de las características que para más información, aconsejamos te dirijas a: Cimex Electrónica. C/ Floridablanca, 54, Ent. 3 A. 08015 Barcelona. Teléfono (93) 224 34 22.



Consultas

P: Les escribo a su sección de cartas para que me aclaren unas dudas que me han surgido con mi CBM 120 (en modo 64):

1) Sé que la colisión de *sprites* con gráficos se realiza con la dirección de memoria 53279; pero lo que no entiendo es cómo saber con qué gráfico determinado ha chocado. Por ejemplo: si un *sprite* choca con una letra A, ¿cómo se puede saber que ha chocado con esa letra y no con otra?

2) Recientemente he adquirido una máquina de escribir electrónica y en las instrucciones dice que se puede conectar como impresora para un ordenador que emplee interface Centronics. ¿Se puede conectar a un ordenador Commodore mediante ese interface?

Gonzalo Aguilera
Avila

R: (1) Como bien sabes, la colisión entre un *Sprite* y un gráfico se lee en la posición 53279, la cual te devuelve un valor que determina qué *sprite* ha colisionado, pero sin especificar qué tipo de carácter gráfico es con el que colisionó.

El Commodore posee 127 caracteres, entre letras, números y símbolos gráficos que pueden presentarse en dos modos, normal e inverso, lo que suma un total de unos 255 tipos de caracteres, con los que puede tropezar un *sprite.*

Tu pregunta ¿cómo se puede saber que ha chocado con esa letra y no con otra?, tiene una complicada respuesta si se pretende solucionar, pues deberías de averiguar por PEEKs en qué situación se produjo la colisión y luego también por PEEK, obtener el código ASCII del carácter con el que colisionó. Esto requiere una laboriosa programación que sólo te recomiendo para un trabajo verdaderamente importante o bonito.

(2) El Commodore es capaz de comunicarse por interface Centronics con una impresora; lo único que pasa, es que tienes que tener ese interface, adquiriéndolo en alguna tienda especializada o fabricándolo tú mismo. En el número 3 de Commodore Magazine se explica, en el artículo «Interfaces para todos», cómo construirse uno.

Templo Maldito



Sugerente es el título con el que Carlos Alonso Losas ha bautizado su juego. El Templo Maldito.

Este programa, para el Vic-20, basa su acción en un famoso templo lleno de pasadizos en cuya cámara más profunda se encuentra una gran piedra de oro.

Convertido en un intrépido aventurero intentarás llegar a la cámara del tesoro, tomar la mayor cantidad de oro posible y volver al lugar de partida. Dispones para ello de 30 segundos y cuatro vidas, que perderás si tocas una de las múltiples calaveras, de personas que perecieron en el intento, si te toca un rayo espectral o si coges más de 25 kg. de la cámara del tesoro.

Para coger el oro debes colocar al hombre encima del tesoro, automáticamente éste se aparta y en el marcador aparecerán 5 kg. (recuerda que sólo se puede hacer esta operación cinco veces, de lo contrario se perderá una vida).

Cada vez que consigas subir el oro al pasadizo superior obtendrás 25 kg. de bonificación y el hombre se situará en el punto de partida.

Si consigues terminar el juego, cosa que lograrás cuando hayas recogido 500 kg. de oro, aparecerá un mensaje de felicitación y la invitación de volver a jugar. En caso contrario se mostrará la puntuación obtenida y la posibilidad de intentarlo otra vez.

El hombre se maneja con las teclas:

Z-Izquierda	X-Derecha
F5-Arriba	F7-Abajo

El listado está dividido en dos partes y corresponden al esquema siguiente:

Primer programa: formación de gráficos.

Segundo programa:

2-5	Inicialización de variables.
8-220	Dibujo del templo
260	Controla el tiempo.
270-280	Dibuja los rayos espectrales
290-320	Movimiento y controlación del hombre.
330	Comprueba si se coge oro.
340	Comprueba si te toca un rayo espectral.
370	Comprueba si se sube el hombre al pasillo superior.
380	Comprueba si se ha ganado.
400-450	Definición de las teclas de movimiento.
500-550	Quita una vida y comprueba si se ha acabado.
600-640	Suma los kg. de oro y comprueba si coges en exceso.
700-750	Datas de los rayos espectrales.
800-850	Rutina de fin de juego si ganas.
1500-1700	Presentación e instrucciones.
2000-2070	Rutina de fin de juego si pierdes.

PROGRAMAS

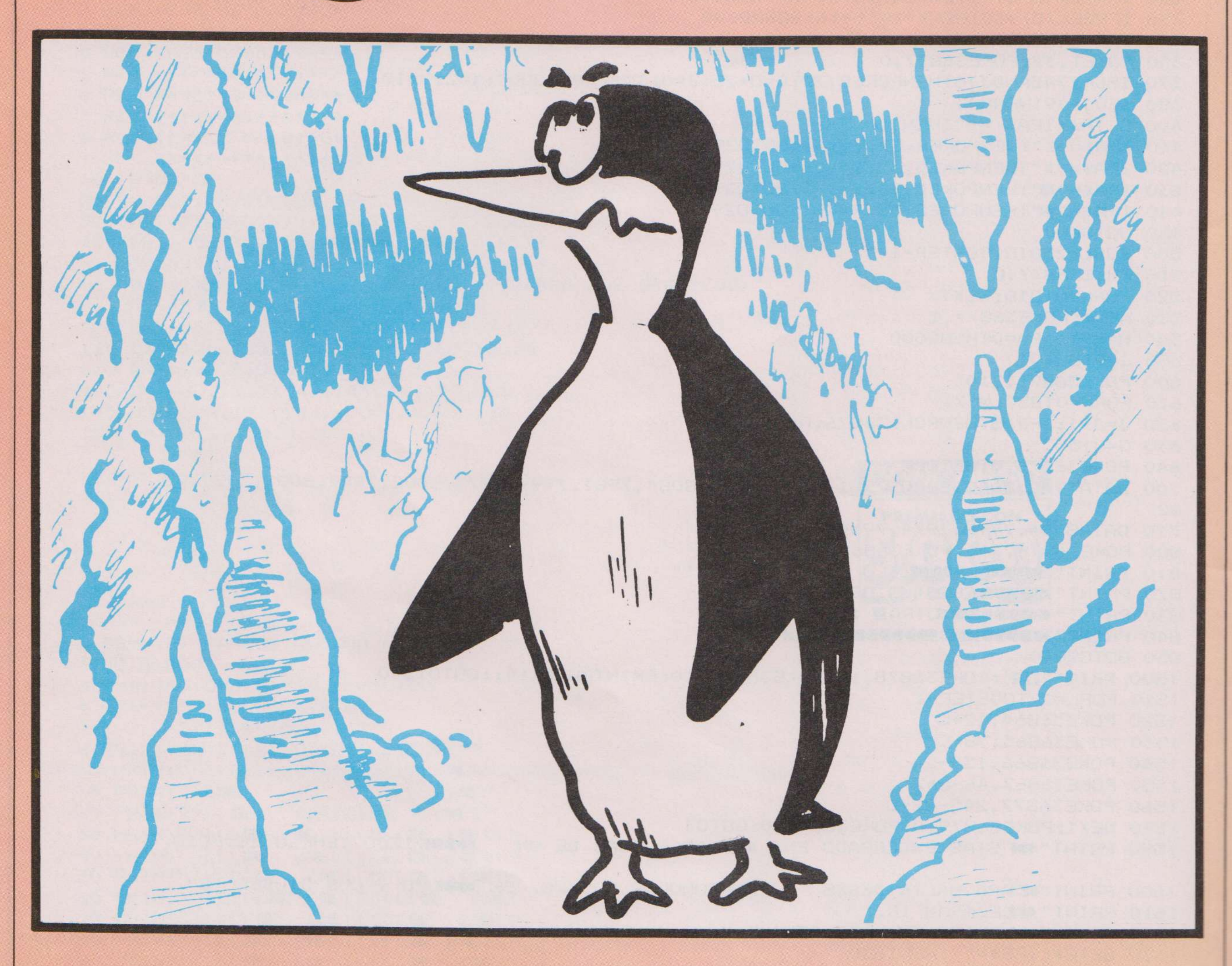
```
1 REM ***************
2 REM *CARLOS ALONSO*
3 REM ***************
4 REM *****LOSAS*****
5 REM ***************
6 REM ***17-9-1986***
7 REM ***************
8 REM TEMPLO MALDITO*
9 REM ***************
10 PRINT"■"
20 POKE52,28:POKE56,28:CLR
30 FORI=7168TO7679:POKEI,PEEK(I+25600):NEXT
40 FORI=7384TO7423:READA:POKEI,A:NEXT
50 FORI=7432TO7439:READA:POKEI,A:NEXT
60 PRINT"■■ESTE PROGRAMA SOLO     CARGA LOS GRAFICOS."
70 PRINT"■VUELVE A CARGAR."
100 DATA251,251,251,0,127,127,127,0
110 DATA126,153,153,255,102,60,36,24
120 DATA28,28,8,62,127,93,20,54
130 DATA33,19,71,47,143,31,127,255
140 DATA129,195,102,60,0,24,60,102
150 DATA3,3,3,0,3,3,3,0
```




```
1 REM (C) CARLOS ALONSO LOSAS.
2 GO!01500
3 H=4:J=0:O=0:P=7680:C=30720
4 PRINT"■":POKE36879,8
5 X=2:Y=17:TI$="000000"
8 PRINT"■■'CCCCCCCCCCCCCCCCCCCCC"
10 PRINT"!CC■££££££££££££££££■CC"
15 PRINT"!C■                   £■C"
20 PRINT"!CC■    ££££££££    £■CC"
30 PRINT"!LLL■ £■LLLLLLLL■ £■LLL"
40 PRINT"!LLL■  £■LLLLLLL■ £■LLL"
50 PRINT"!CCCC■  £■CCCCCC■  £■CC"
60 PRINT"!CCCCC■ £■CCCCCCC■  £■C"
70 PRINT"!CCCC■  £■CCCCCC■  £■CC"
80 PRINT"!CCC■  £■CCCCCCC■ £■CCC"
90 PRINT"!CCC■  £■CCCCCCC■  £■CC"
100 PRINT"!CCCC■  £■CC   CCC■  £■C"
110 PRINT"!CCCC■  £■CCCCCC■  £■CC"
120 PRINT"!CCCCC■  ££££££  £■CCC"
130 PRINT"!CCCCC■              £■CCC"
140 PRINT"!CCCC■    ££££££££■CCCC"
150 PRINT"!CCC■    £■CCCCCCCCCCCCC"
160 PRINT"!CCC■     ££££££££■CCCC"
170 PRINT"!CCC■■                £■CCC"
180 PRINT"!CCCC■£££■             £■CC"
190 PRINT"!■KILOS■CC■££££££££■■CCC"
200 PRINT"!■DE■CCCCCCCCCCCCCCCCCC"
210 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■"MID$(TI$,5,2)
220 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■! ■ORO";O;" HOMBRES"H"■"
260 IFTI$="000030"THEN2000
270 READT:IFT=7960THENRESTORE
280 POKET,31:POKET+C,5:POKE36877,250
290 D=P+X*22+Y
300 POKED,29:POKED+C,1:POKE8115,30:POKE8115+C,7
```

```
310 IFPEEK(D-22)=27ORPEEK(D+22)=27THEN500
320 IFPEEK(D-1)=27ORPEEK(D+1)=27THEN500
330 IFPEEK(D)=30THENX=19:Y=16:GOSUB600
340 IFD=TTHEN500
350 POKET,32:POKE36877,0
370 IFD<7745ANDJ>0THENPOKED,32:O=O+25:J=0:TI$="000000":X=2:Y=17
380 IFO>499THEN800
400 GETA$:IFA$ =""THEN210
410 IFA$="Z"THENPOKED,32:Y=Y-1:GOTO290
420 IFA$="X"THENPOKED,32:Y=Y+1:GOTO290
430 IFA$="■"THENPOKED,32:X=X-1:GOTO290
440 IFA$="■"THENPOKED,32:X=X+1:GOTO290
450 GOTO400
500 FORL=250TO190STEP-1
510 POKE36877,L
520 FORW=0TO10:NEXT
530 NEXT:POKE36877,0
540 H=H-1:IFH=0THEN2000
550 J=0:GOTO4
600 POKE36876,210
610 FORW=0TO50:NEXT
620 J=J+1:IFJ>5THENPOKE36876,0:GOTO500
630 O=O+5
640 POKE36876,0:RETURN
700 DATA7982,8062,8000,7951,8003,7996,8084,7981,7999,7973,8002,7997,8001,7929,80
62
710 DATA8084,7998,7974,7960
800 POKE36879,216:POKE36869,240
810 PRINT"■■■■■■■■■■!!LO CONSEGUISTE!!"
820 PRINT"■■■■■■ERES UN GENIO"
830 PRINT"■■■■■■■■OTRA■ ■PARTIDA■"
840 PRINT"■■■■■■■S■■■■■■■■■■■■N■"
850 GOTO2040
1500 PRINT"■■":POKE36878,15:POKE36879,10:PRINTCHR$(14):GOTO1590
1510 FORL=22TO0STEP-1
1520 POKE36864,12+L
1530 POKE36865,38+L
1540 POKE36866,150-L
1550 POKE36867,46-L*2
1560 POKE36877,245-22+L
1570 NEXT:POKE36879,8:POKE36877,0:GOTO3
1590 PRINT"■■■STAS ENCERRADO EN  ■■■LAS RUINAS DE UN   ■■■■VIEJO TEMPLO EGIPCIO.
"
1600 PRINT"■■ARA SALIR DEBES     ■■ACUMULAR 500 KG. DE ■■■■RO Y LOS DIOSES TE"
1610 PRINT"■■DEJARAN IR."
1620 PRINT"■■■■ULSA UNA TECLA■"
1630 GETE$:IFE$=""THEN1630
1640 PRINT"■■■■■L HOMBRE QUE MANEJAS ■■E MUEVE CON:"
1650 PRINT"■■■=■ZQUIERDA            ■■=■ERECHA "
1660 PRINT"■■■5=■RRIBA              ■■7=■BAJO"
1670 PRINT"■■■■■!!■UENA SUERTE!!■    ■ ■ CUIDADO,HAY MUCHOS  ■PELIGROS."
1680 PRINT"■■■■■■■ULSA UNA TECLA■"
1690 GETE$:IFE$=""THEN1690
1700 PRINT"■":POKE36869,255:GOTO1510
1900 POKE36877,0:STOP
2000 POKE36869,240:POKE36879,152
2010 PRINT"■■■■■■■■■■■KG. DE ORO ■";O
2015 PRINT"■■■■■■■■■■■O SIENTO"
2016 PRINT"■■■■■■■■OTRA VEZ SERA"
2020 PRINT"■■■■■■■■■OTRA■ ■PARTIDA■"
2030 PRINT"■■■■■■■S■■■■■■■■■■■■N■"
2040 GETS$:IFS$=""THEN2040
2050 IFS$="S"THENPOKE36869,255:PRINT"■":GOTO1510
2060 IFS$="N"THENPOKE36869,240:GOTO1500
2070 GOTO2040
```

Pengo



Sencillo juego el que nos propone José Manuel Arroyo para el Vic-20 sin ningún tipo de ampliación. Su nombre es Pengo y aunque realmente no destaca por sus gráficos, resulta bastante entretenido.

El juego en sí consiste en guiar a un pequeño pingüino, situado en el centro de la pantalla, a través de estrechos y aleatorios laberintos creados con cubitos de hielo. Tu objetivo es comerte un mínimo de 10 estrellas de un máximo de 20, distribuidas por toda la pantalla.

Esto lo deberás realizar lo más rápidamente posible, pues sólo dispones de dos minutos y hay cuatro monstruitos que interceptarán tu camino, llegando a destruirte. Estos engorrosos monstruos tienen la facultad de atravesar los cubos de hielo y de comerse las estrellas.

Una vez superada esta etapa, pasarás a la siguiente pantalla, siempre diferente y cada vez con más cubitos de hielo.

El pingüino se maneja con las sigientes teclas: W-arriba, X-abajo, A-izquierda y D-derecha.

El listado se encuentra dividido en dos partes, que tendrás que cargar y ejecutar por separado. En la primera parte se cargan los datas de los gráficos y en la segunda el juego propiamente dicho. □




```
10 POKE52,28:POKE56,28
20 FORT=7168TO7215:READA:POKET,A:NEXT
30 DATA0,0,0,0,0,0,0,0
40 DATA24,36,36,24,126,90,24,36
50 DATA0,84,56,124,56,84,0,0
60 DATA24,60,231,60,24,255,60,66
70 DATA126,129,129,129,161,177,129,126
80 DATA162,145,138,82,0,137,223,126
```

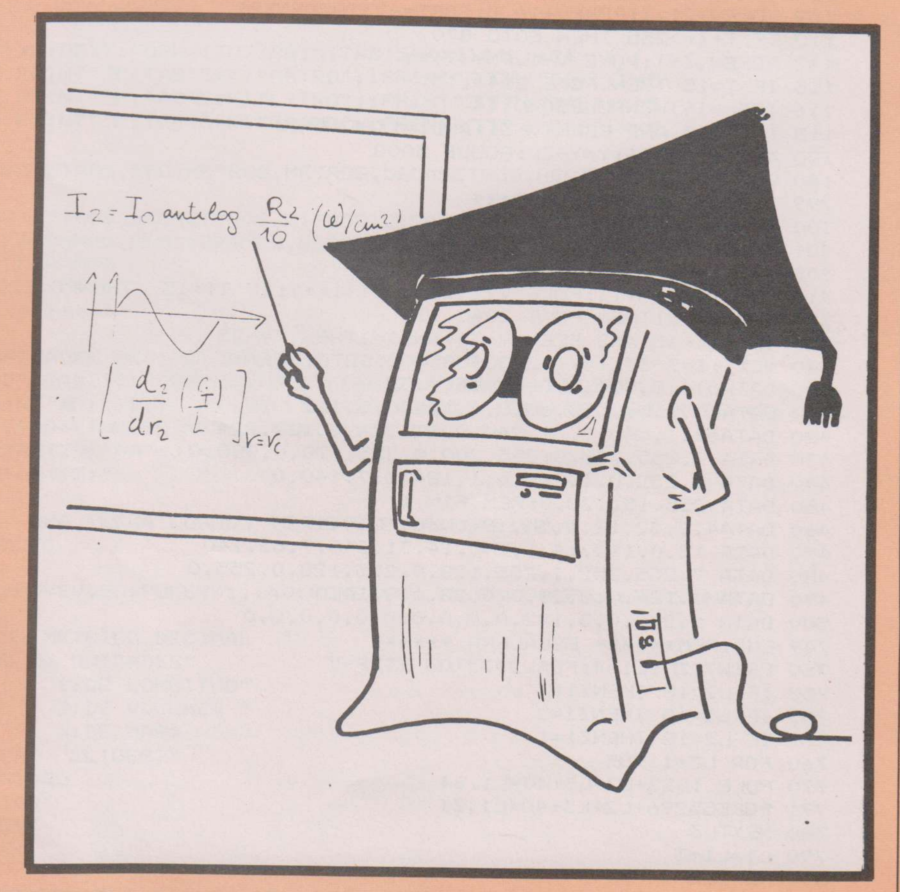
```
0 POKE36879,106:POKE36878,15:POKE650,128:PRINT""
10 FORT=46TO0STEP-2:POKE36883,T:FORTT=1TO50:NEXTTT:NEXTT
20 PRINT" PENGO":PRINT"  PENGO":PRINT"   PENGO"
30 PRINT"    PENGO":PRINT"     PENGO":PRINT"      PENGO"
40 PRINT"       PENGO":PRINT" POR J.M.ARROYO.":PRINT"    P E N G O ."
50 FORT=0TO46STEP2:POKE36883,T:FORTT=1TO50:NEXTTT:NEXTT
60 FORT=1TO5000:NEXT:PRINT""
70 POKE36869,255:W=160:X=0:K=0:S=7910:S1=38630:V=3:N=0:L=1
80 POKE36879,30:R1=7702:R2=7723:R3=8142:R4=8163:K1=0:K2=0:K3=0:K4=0:X1=0:X2=0:X3
=0:X4=0
90 T1=38422:T2=38443:T3=38862:T4=38883:H=0
100 FORT=1TOW:W1=INT(0+462*RND(0)):POKE7702+W1,4:POKE38422+W1,6:NEXT
110 FORT=1TO20:W1=INT(0+462*RND(0)):POKE7702+W1,2:POKE38422+W1,4:NEXT
120 TI$="000000"
150 PRINT""H;N""
160 Q=Q+L:IFQ>100THENL=-1
170 IFQ<0THENL=1
180 POKE36876,150+Q
185 POKEO1,0
190 A=INT(1+4*RND(0)):IFA=1THENX1=X1+1:K1=1
200 IFA=2THENX1=X1-1:K1=-1
210 IFA=3THENX1=X1+22:K1=22
220 IFA=4THENX1=X1-22:K1=-22
225 O1=R1+X1
230 IFO1<7702ORO1>8163THENX1=X1-K1
235 POKEO1,3:POKET1+X1,2:POKEO2,0
240 B=INT(1+4*RND(0)):IFB=1THENX2=X2+1:K2=1
250 IFB=2THENX2=X2-1:K2=-1
260 IFB=3THENX2=X2+22:K2=22
```

```
270 IFB=4THENX2=X2-22:K2=-22
275 O2=R2+X2
280 IFO2<7702ORO2>8163THENX2=X2-K2
285 POKEO2,3:POKET2+X2,2:POKEO3,0
290 C=INT(1+4*RND(0)):IFC=1THENX3=X3+1:K3=1
300 IFC=2THENX3=X3-1:K3=-1
310 IFC=3THENX3=X3+22:K3=22
320 IFC=4THENX3=X3-22:K3=-22
325 O3=R3+X3
330 IFO3<7702ORO3>8163THENX3=X3-K3
335 POKEO3,3:POKET3+X3,2:POKEO4,0
340 D=INT(1+4*RND(0)):IFD=1THENX4=X4+1:K4=1
350 IFD=2THENX4=X4-1:K4=-1
360 IFD=3THENX4=X4+22:K4=22
370 IFD=4THENX4=X4-22:K4=-22
375 O4=R4+X4
380 IFO4<7702ORO4>8163THENX4=X4-K4
400 POKEO4,3:POKET4+X4,2
402 GETA$:POKES+X,0
403 IFA$="A"THENX=X-1:K=-1
404 IFA$="D"THENX=X+1:K=1
405 IFA$="W"THENX=X-22:K=-22
406 IFA$="X"THENX=X+22:K=22
410 IFPEEK(S+X)=4THENX=X-K
420 IFPEEK(S+X)=2THENGOSUB540
430 IFPEEK(S+X)=3THENGOSUB560
440 IFS+X<7702ORS+X>8163THENX=X-K
450 POKES+X,1:POKES1+X,0
460 IFPEEK(R1+X1)=1THENGOSUB560
470 IFPEEK(R2+X2)=1THENGOSUB560
480 IFPEEK(R3+X3)=1THENGOSUB560
490 IFPEEK(R4+X4)=1THENGOSUB560
500 IFH=>10THEN610
505 IFTI$>"000200"ANDH<10THEN610
510 GOTO150
540 POKE36876,0:H=H+1:POKE36876,245:FORT=1TO15:NEXT
550 POKE36876,0:N=N+10:RETURN
560 POKES+X,5:POKES1+X,0:POKE36876,0
570 V=V-1:FORT=1TO200:FM=INT(1+100*RND(0)):POKE36876,150+FM:NEXT
580 POKE36876,0:X=0:K=0
590 IFV=0THEN700
600 RETURN
610 POKE36869,240:PRINT"▯▮":POKE36879,106:POKE36876,0
620 IFTI$<"000030"THENN=N+2000
630 IFTI$<"000100"THENN=N+1000
640 IFTI$<"000130"THENN=N+500
650 PRINT"▮ TIEMPO<1/2M...2000PTS":PRINT"▮ TIEMPO<1M.....1000PTS"
660 PRINT"▮ TIEMPO<1Y1/2M.500PTS":PRINT"▮▮ TIEMPO>2M.....0PTS"
670 PRINT"▮▮ PUNTOS:"N:FORT=1TO5000:NEXT
680 W=W+20:IFW>360THENW=360
690 POKE36869,255:POKE36879,30:H=0:X=0:K=0:GOTO100
700 POKE36869,240:PRINT"▯■":PRINT"▮ PUNTOS:"N:POKE36876,0
710 PRINT"▮▮▮▮▮▮▮ FIN DEL JUEGO."
720 PRINT"▮▮▮▮      PULSA LA 'S'"
730 GETA$:IFA$<>"S"THEN730
740 GOTO0
```

Enseñanza informatizada

Dos programas de Juan Alvarez Avello han conseguido superar la selección de concurso. Desde Madrid y con fines pedagógicos, que los ordenadores no pueden dedicarse exclusivamente a matar todo tipo de marcianitos, y que quizá los programadores se deberían fijar más en otros campos como pueden ser los de la enseñanza.

Es por esto por lo que se dirige a nosotros adjuntándonos los programas, el Sistema Métrico Decimal para alumnos de 4.° de EGB y la Representación de Funciones para alumnos de 2.° de BUP, programas que han sido utilizados en el aula de su colegio con notable éxito.



Con ello pretende dar una muestra de lo que hacen de informática en algunos centros y para que pueda servir de estímulo y ejemplo para otros, a fin de implantar las llamadas «Nuevas Tecnologías» en la enseñanza moderna.

— El primero, Sistema Métrico Decimal, está pensado para alumnos de 4.° de EGB que deben pasar unidades de longitud, masa y volumen en otros múltiplos y submúltiplos suyos. Para ello, el alumno puede ayudarse de una «escalera» por la que puede subir y bajar un canguro. En cada escalón está un múltiplo y submúltiplo y el niño debe dirigirse de un peldaño a otro para multiplicar o dividir por la unidad seguida de ceros.

Al principio del programa se dan una serie de tablas y equivalencias y a continuación se hacen 10 preguntas al alumno pudiéndose éste ayudar del canguro subiéndolo o bajándolo hasta llegar al escalón indicado. Para subir se utiliza la tecla F-1 y para bajar la F-7. Cuando se alcanza el escalón preguntado se debe multiplicar o dividir por la unidad seguida de ceros según el número de saltos que el canguro haya dado. Al responder se debe apretar primero a la barra espaciadora y después apretar RETURN. Al finalizar te da la puntuación y una observación general sobre lo hecho.

— El segundo, Funciones, representa y superpone funciones ampliables a escala, calcula máximos y mínimos por el método de Darboux, obteniéndose al final un intervalo donde se encuentra el máximo o el mínimo. El programa calcula también integrales definidas de áreas por el método de los rectángulos. Para borrar la pantalla de alta resolución hay una subrutina en código de máquina.

```
10 REM SISTEMA METRICO DECIMAL
20 REM POR JUAN DANIEL ALVAREZ AVELLO
30 REM       COMMODORE 64
40 V=53248:A=1024:PRINT"□":Z=44:Y=165
50 FOR L=0TO63:READ A:POKE832+L,A:NEXT
60 FOR L=0TO63:READ A:POKE896+L,A:NEXT:GOSUB3000:LA=1:GOSUB1000
70 GOSUB750: POKE V+16,0:POKE 2040,13:POKEV+21,1
75 REM POKE53280,  :POKE53281,
80 POKE V+23,1:POKEV+29,1:POKEV+39,12
90 DEFFNY(X)= .1*X↑2:GOTO 1020:END
95 POKEV,Z+5-23:POKE V+1,Y+FNY(-25)
109 POKEV+21,1:POKEV+16,0:FORT=-23TO20STEP2
110 IF T+Z=>255 THEN GOTO 870
112 POKEV,Z+T:POKE V+1,FNY(T)+Y
115 IF T=15 THEN POKE 2040,13
116 IFT=-19THENPOKE2040,14
118 IF T>15 AND PEEK(V+31)AND1=1THENT=20
120 NEXT:Z=Z+36:Y=Y-20:GOSUB 5000
140 GOTO10066
299 REM*****HACIA ABAJO***
300 X=-25:P=19:SM=40
301 POKEV+21.1:POKEV+16.0
305 FORT1=P  TOX STEP-2
310 POKEV.Z-SM+T1:POKE V+1,24+FNY(T1)+Y:IF T1=13  THENPOKE2040,14
320 IF T1=-21THEN POKE 2040.13
330 IF T1<-15 AND PEEK(V+31)AND1=1THEN T=-25
340 NEXT:Z=Z-37:Y=Y+22:GOSUB5000:GOTO10066
400 DATA0.0.0.0.0.0.0.0.0.0.0.32
410 DATA 0.0.56.0.0.60.0.0.126.0.0.112
420 DATA 0.1,240.0.31.240.0.63.224,0.127,224
430 DATA 0.255.248,0.255.200.0.255.128,0.255,0
440 DATA 1.252.0.1,248,0.1,184.0.7.140,0
450 DATA 255.15.128.0:REM FIN 1
460 DATA4.0,32.12,0,56.12.0.60.12.0.127
470 DATA 12.0.112.14.1,240.14.31.240.7,63,240
480 DATA 7.255.252,1,255,128.0.255.128,0,255,0
490 DATA 0.126.0,0,28.0,0.28,0.0.14,0
500 DATA 0.2,0,0,3,192,0,0.0.0,0,0,0,0,0
749 END:REM****** ESCALERA ******
750 PRINT"□":L1=4:FORL2=33TO3 STEP-5
752 IF L2>18 THENZ1=7
755 IF L2=18 THENZ1=2
757 IF L2<18 THENZ1=1
760 FOR L3=1 TO5
770 POKE 1023+L2+L3+40*L1.64
771 POKE55296+L2+L3+40*L1,Z1
780 NEXTL3
790 L1=L1+3
800 NEXTL2
810 Y3=4:Y1=33:FOR Y2=3 TO21STEP 3
811 IF Y2<10 THENZ2=7
812 IF Y2=12 THENZ2=2
813 IF Y2>12 THENZ2=1
815 POKE 1024+Y1+40*(Y2+Y3-3),112
816 POKE55296+Y1+40*(Y2+Y3-3),Z2
820 FOR Y3=2TO3
830 POKE 1024+Y1+40*(Y2+Y3),66
831 POKE55296+Y1+40*(Y2+Y3),Z2
840 NEXT Y3
844 POKE 1024+Y1+40*(Y2+Y3),125
845 POKE55296+Y1+40*(Y2+Y3),Z2
850 Y1=Y1-5:NEXTY2
860 PRINT"▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮";TAB(21);"SALTOS :":GOSUB5000:PRINT"▤":RETURN
870 REM**** BYTE MAS  SIGNIFICATIVO****
880 POKEV+16,1
890 Z=0:Y=20
900 FOR Q=10 TO 18 STEP2
```

```
910 POKEV,Q-10:POKEV+1,Y+FNY(Q)-10
920 IF Q=16THENPOKE2040,13
930 NEXT:GOSUB5000:GOTO10066
940 REM ****** HACIA ABAJO*******
945 Y=10:Z=0:POKEV+21,1:POKEV+16,1
950 FOR Q =18TO10 STEP-2
960 POKEV,Q-10:POKEV+1,  FNY(Q)
965 IF Q=12 THENPOKE2040,13
970 NEXT:Z=299:Y=-12:X=-23:SM=50:P=1:GOTO301
1000 FORTT=1TO F:FOR P=1 TO7:READ B$:J$(P)=B$:J1$(P)=LEFT$(J$(P),1):NEXTP:NEXT
1004 C$=J1$(4)
1011 IFLA=1THEN 2000
1020 FOR G1=1 TO11:PRINT"▮":NEXT:FORK=1TO7:PRINTTAB(5*K)J1$(K);C$
1030 IF K=4 AND F=1 THEN PRINT"┓";TAB(5*K)"METRO";:PRINT"┳┳┓";"DAM":K=5
1033 IF K=4 AND F=2 THEN PRINT"┓";TAB(5*K)"LITRO";:PRINT"┳┳┓";"DAL":K=5
1035 IF K=4 AND F=3 THEN PRINT"┓";TAB(5*K)"GRAMO";:PRINT"┳┳┓";"DAG":K=5
1040 PRINT"┳┳┳┳┳┓":NEXT
1045 DATA MILIMETROS,CENTIMETROS,DECIMETROS,METROS,DECAMETROS,HECTOMETROS
1047 DATA KILOMETROS
1050 DATA MILILITROS,CENTILITROS,DECILITROS,LITROS,DECALITROS,HECTOLITROS
1060 DATA  KILOLITROS,MILIGRAMOS,CENTIGRAMOS,DECIGRAMOS,GRAMOS,DECAGRAMOS
1070 DATA HECTOGRAMOS,KILOGRAMOS
1999 GOTO10000
2000 REM******* CUADROS*********
2001 PRINT"┓"
2010 PRINT"▮▬          UNIDADES DE";D$;"▮▮▮"
2020 PRINTTAB(5)"UNIDADAD";TAB(15)"ABREVIATURA";TAB(27)"EQUIVALENCIAS"
2030 FORIO=7TO1STEP-1:PRINT"▮";TAB(5) J$(IO);TAB(20)J1$(IO);C$;TAB(30)10↑(IO-4);
2040 PRINTTAB(38)J1$(4):IFIO=1THENPRINT"┓";TAB(30)" .001   ";
2050 IFIO=5THENPRINT"┓";TAB(20)"DA";J1$(4)
2060 IFIO=4THENPRINT"┓";TAB(15)"       "
2070 NEXT
2080 PRINT:PRINT"▮▮PULSA  UNA TECLA CUANDO TE LO SEPAS":POKE198,0
2090 GETRR$:IFRR$=""THEN2090
2100 RETURN
3000 PRINT"┓":POKEV+21,1:POKEV,80:POKEV+1,140:POKE2040,13:POKEV+23,1:POKEV+29,1
3001 POKEV+16,0:POKEV+39,13
3005 PRINT"▮       SISTEMA METRICO DECIMAL  "
3010 PRINT"▮▮▮▮     TIPOS DE UNIDADES"
3020 PRINT"▮▮              1:DE LONGITUD"
3030 PRINT"▮▮              2:DE VOLUMEN "
3040 PRINT"▮▮              3:DE MASA  "
3050 PRINT"▮▮▮▮▮ QUE UNIDAD  ELIGES?"
3060 GETF:IFF=0 ORF>3THEN3060
3070 IFF=1THEND$=" LONGITUD"
3080 IFF=2THEND$=" VOLUMEN"
3090 IFF=3THEND$=" MASA "
3100 POKE V+21,0:RETURN
5000 PRINT"S▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮";TAB(30);SL;"S▮▮";TAB(28):RETURN
10000 CW=CW+1:C=INT(RND(1)*1000)+1
10010 R9=INT(RND(1)*7)+1
10020 S=INT(7*RND(2))+1:
10030 IF R9=S THEN 10020
10040 R$=J$(R9)
10050 S$=J$(S):Z=(40*(R9-1)+67):Y=24*(5-R9)+30:GOSUB 5000
10055 PRINT"S▮             PREGUNTA NUMERO ";CW
10060 PRINT"S▮▬       SI TENEMOS";C;R$
10061 PRINT"S▮▮    CUANTOS ";S$;" SON";
10062 IF R9=7THEN POKEV+16,1:POKEV+21,1:POKEV,10:POKEV+1,46
10065 IFR9<7THENPOKEV+21,1:POKEV+16,0:POKEV,Z+5-19:POKEV+1,Y+FNY(-25):POKE198,0
10066 GETA$:IFA$=""THEN10066
10067 IFA$="■" AND Z<>0 THEN SL=SL+1:GOTO 109
10068 IFA$="▮" ANDZ<>0ANDZ<>307 ANDZ>77THEN SL=SL+1:GOTO 300
10069 IFA$=" "THEN INPUTG$:G=VAL(G$):GOTO10076
10070 IFA$="▮" ANDZ=0ORZ=307 THEN SL=SL+1:GOTO 945
10075 GOTO 10066
```

```
10076 K6=C*10↑(R9-S):K9$=STR$(K6):K=VAL(K9$)
10080 IF  G=K THEN PRINT"■■■■■      CORRECTO":CN=CN+1:GOTO10095
10090 IFK <>G THEN PRINT"ERROR LA RESPUESTA ERA"; K
10095 FORW3=0TO3000:NEXT:POKE198,0:SL=0
10100 PRINT"■                                                   ";
10102 PRINT"                                                    ";
10104 PRINT"                                                   ";
10105 IF CW<10 THEN  10000
10107 POKE V+21,0
10110 PRINT"■":PRINT"■■■■■■■■■■          HAS TENIDO UN ";CN
10120 IFCN<5THENPRINT"■■■      DEBES REPASARTELO OTRA VEZ":FORT=1TO2000:NEXT:RUN
10130 IF CN<7THENPRINT"■■■       DEBERIAS ESTUDIARLO MEJOR":GOTO10200
10140 PRINT"■■■■CREO QUE TE LO SABES BIEN ENHORABUENA"
10200 PRINT"■■■■■■■■      QUIERES SEGUIR REPASANDO?(S/N)"
10210 GET Y$:IF Y$=""THEN 10210
10220 IF Y$="S"THEN RUN
10230 IF Y$<>"N"THEN 10210
10250 END
```

```
10 REM REPRESENTACION DE FUNCIONES
20 REM POR JUAN DANIEL ALVAREZ
30 REM COMMODORE 64
40 PRINT"■■":GOSUB 1100
50 REM POKE53280, :POKE53281,
100 GOSUB 1200:ON Z GOTO 120,115,120,120
110 GOTO 100
115 IFGH=0THEN110
120 GOTO 1000
130 DEFFNY(X)=:DEFFNB(X)=
131 PRINT"■"
133 IF Z=3THEN2000
134 IF Z=4THEN4000
135 GOSUB1130:IFZ=1THENGOSUB1400
136 IF Z=3THEN2050
140 PA=1:GOSUB 1300:GOSUB1500:GOSUB1355: GOTO100
1000 REM ENTRA FUNCION
1003 PRINT"■■■         ■";:PRINTRIGHT$(A$(Z),LEN(A$(Z))-2):PRINT"■"
1005 PRINT"■■      - FUNCION RACIONAL(S/N)"
1007 GETW$:IFW$=""THEN1007
1010 IF W$="S"THEN 1080
1015 IF W$="N"THEN 1030
1017 GOTO1005
1027 PRINT"■■■          ■  ■";:PRINTRIGHT$(A$(Z),LEN(A$(Z))-2):PRINT"■"
1030 INPUT"■■■  ENTRE FUNCION DE VARIABLE X ";A$
1040 INPUT"■■■■■      ESCALA DE AMPLIACION(20)";ES :PT=1
1050 PRINT"■■130 DEFFNY(X)=";A$;":DEFFNB(X)=1":PRINT"Z=";Z":ES=";ES;":GOTO130"
1060 GOTO1095
1075 PRINT"■■■          ■  ■";:PRINTRIGHT$(A$(Z),LEN(A$(Z))-2):PRINT"■"
1080 INPUT"■■■   - ENTRE EL NUMERADOR ";A$
1085 INPUT"■■   - ENTRE EL DENOMINADOR ";J$:
1090 INPUT"■■■    ESCALA DE AMPLIACION(20) ";ES
1093 PRINT"■■130 DEFFNY(X)=";A$;":DEFFNB(X)=";J$:PRINT"Z=";Z":ES=";ES;":GOTO130"
1095 POKE631,19:POKE632,13:POKE633,13:POKE198,3 :END
1100 REM LIMPIA HI-RES
1110 FOR X=0TO77:READA:POKE49152+X,A:NEXT:RETURN
1130 POKE53272,PEEK(53272)OR8:POKE53265,PEEK(53265)OR32
1135 IF Z=2THEN FORA=0TO999:POKE1024+A,13:NEXT :RETURN
1140 SYS49152:RETURN
1200 REM MENU
```

```
1210 A$(0)="          MENU":A$(1)="1-NUEVA FUNCION":A$(3)="3-MAXIMOS Y MINIMOS"
1215 A$(2)="2-SUPERPOSICION DE FUNCIONES"
1217 A$(4)="4-INTEGRAL DEFINIDA"
1220 PRINT"■■■■":FORA=0TO 4:PRINT"          ";A$(A);"■■■":NEXT
1230 PRINT"QUE OPCION ELIGES?"
1240 GETZ$:IFZ$=""THEN1240
1250 Z=VAL(Z$):IFZ=0THEN1240
1260 RETURN
1300 REM REPRESENTA LA FUNCION
1310 FOR X=0TO319:X1=(-159+X)/ES
1315 Y1=FNY(X1):Y2=FNB(X1):IFY2=0THENNEXT:RETURN
1317 Y=Y1/Y2:Y=-Y*ES+100:IFY>=198ORY<=0THENNEXT:RETURN
1320 BY=8192+(INT(Y/8)*320)+(INT(X/8)*8)+(YAND7)
1330 BI=7-(XAND7):R=PEEK(BY)OR2↑BI:
1340 POKEBY,R
1350 NEXTX:RETURN
1355 POKE198,0
1360 GETR$:IFR$=""THEN1360
1370 POKE53265,PEEK(53265)AND223:POKE53272,PEEK(53272)AND247
1380 GH=1+GH:RETURN
1400 REM EJES
1410 FORY=0 TO  7840 STEP 320:FORX=0TO 7 STEP4:POKE8344+X+Y,1 :NEXT:NEXT
1420 FORY=0TO319STEP8:POKE12036+Y,133:NEXT:RETURN
1500 REM NUMEROS
1510 CN=4:FORJ=1TO20:FORK=0TO7:CN=CN+1
1520 IF CN=ES THEN POKE8192+4000-K-1-320*J,7:POKE12192-K-1-320*J+8,192
1525 IF CN=ES THEN POKE8192+4000+K+320*J,192:POKE12192+K+320*J-8,7:CN=0
1530 NEXT:NEXT
1540 REM HORIZONTAL
1550 M=-1:FORJ=0TO100:FORK=0TO7:M=M+1
1560 IFM=ESTHENM=-1:FORI=-1TO1:POKE12196+J*8+I,PEEK(12196+J*8+I)OR2↑(7-K):NEXT
1563 IF12196+J*8>12348 THEN1580
1570 NEXT:NEXT
1580 M=-1:FORJ=1TO100:FORK=0TO7:M=M+1
1590 IFM=ESTHENM=-1:FORI=-1TO1:POKE12196-J*8+I,PEEK(12196-J*8+I)OR2↑K:NEXT
1593 IF12196-J*8<12044 THEN RETURN
1595 NEXT:NEXT :RETURN
2000 REM MAXIMOS Y MINIMOS
2010 PRINT"■":PRINT"■    CALCULANDO...■■"
2050 FOR E=-159/ES TO 159/ES STEP 3.8
2060 A1=E:B1=E+4 :
2070 A=FNY(A1)/FNB(A1):B=FNY(B1)/FNB(B1)
2080 IFA<B AND ABS(A-B)>.0001 THEN A1=(A1+B1)/2:GOTO2070
2090 IFB<=A AND ABS(A-B)>.0001 THEN B1=(A1+B1)/2 :GOTO2070
2100 IF ABS(A-B)<.01 AND A1=E THEN NEXT:GOTO2120
2105 IF ABS(A-B)<.01 AND B1=E+4THEN NEXT:GOTO2120
2110 IFABS(B-A)<.01 AND A<>E AND B<>E+4ANDB1<159/ESTHEN GOSUB 2200:NEXT:GOTO2500
2120 IFXX<>1THEN PRINT"NO TIENE MAXIMOS"
2199 GOTO2500
2200 PRINT"■MAXIMO ■EN EL INTERVALO";A1;B1:PRINT:XX=1:RETURN
2210 PRINT"■MINIMO ■EN EL INTERVALO ";A1;B1:PRINT:XX=1:RETURN
2500 REM MINIMOS
2505 :PRINT"----------------------------------------■"
2510 :FOR  E=-159/ES TO 159/ES STEP 3.5
2520 A1=E:B1=E+4 :
2530 A=FNY(A1)/FNB(A1):B=FNY(B1)/FNB(B1)
2540 IFA<B AND ABS(A-B)>.0001 THEN B1=(A1+B1)/2:GOTO2530
2550 IFB<=A AND ABS(A-B)>.0001 THEN A1=(A1+B1)/2 :GOTO2530
2560 IF ABS(A-B)<.01 AND A1=E THEN NEXT: GOTO2590
2570 IF ABS(A-B)<.01 AND B1=E+4THEN NEXT:GOTO2590
2580 IFABS(B-A)<.01 AND A<>E AND B<>E+4ANDB1<159/ESTHEN GOSUB 2210:NEXT:
2590 IFXX<>1THENPRINT"NO TIENE MINIMOS"
2600 POKE198,0 :PRINT"■■■■■          PULSE UNA TECLA"
2610 GET R$:IFR$=""THEN2610
2620 GOTO 100
3000 DATA173,24,208,9,8,141,24,208,173,17,208
```

PROGRAMAS

```
3010 DATA 9,32,141,17,208,169,32,133,252,169,0
3020 DATA 133,251,162,31,160,0,145,251,136,208,251
3030 DATA 230,252,202,208,244,160,63,145,251,136,16,251
3040 DATA 169,0,133,253,169,4,133,254,169,3,170
3050 DATA 160,0,145,253,136,208,251,230,254,202,208
3060 DATA 244,160,0,145,253,200,192,232,208,249,96
4000 REM INTEGRAL DEFINIDA
4005 PRINT"■■■      ■■ INTEGRAL DEFINIDA "
4010 PRINT"■■■■   CALCULO DEL AREA COMPRENDIDA ENTRE "
4015 PRINT"■     EL EJE X Y LA FUNCION "
4020 PRINT"■   ENTRE DOS VALORES A Y B"
4030 INPUT"■■    VALOR DE A";V
4040 INPUT"■    VALOR DE B";W
4050 IFW<=VTHENPRINT"■■■■   B DEBE SER MAYOR QUE A":FORLL=1TO1000:NEXT:GOTO4000
4060 INPUT"■■■ EN CUANTOS PASOS POR CADA UNIDAD ";F :IF F=<0THEN4060
4065 OP=(W-V)/F
4070 FOR K=V TO W STEP OP
4080 IFFNB(K)=0THENPRINT"■■■■■      AREA INFINITA AL HABER UNA ASINTOTA":GOTO4130
4090 JH=OP*FNY(K)/FNB(K)
4100 JI=JI+JH:NEXT:
4105 PRINT"■■■      ■■ INTEGRAL DEFINIDA "
4110 PRINT"■■■■ EL AREA TOTAL ES ";JI;"U.S."
4130 PRINT"■■■■■■           PULSE UNA TECLA":POKE198,0
4140 GETR$:IFR$=""THEN4140
4150 GOTO100
```

CODIGOS DE CONTROL PARA EL VIC-20 Y EL C-64

Colores del VIC-20 y del 64

Cómo se ve	Cómo se teclea	Efecto conseguido
	Ctrl + 1	Negro
	Ctrl + 2	Blanco
	Ctrl + 3	Rojo
	Ctrl + 4	Cian
	Ctrl + 5	Púrpura
	Ctrl + 6	Verde
	Ctrl + 7	Azul
	Ctrl + 8	Amarillo

Colores del 64 solamente

Cómo se ve	Cómo se teclea	Efecto conseguido
	Cbm + 1	Naranja
	Cbm + 2	Marrón
	Cbm + 3	Rosa
	Cbm + 4	Gris oscuro
	Cbm + 5	Gris medio
	Cbm + 6	Verde claro
	Cbm + 7	Azul claro
	Cbm + 8	Gris claro

Códigos de cursor y control

Cómo se ve	Cómo se teclea	Efecto conseguido
	Home	Cursor a casa
	Shift + home	Limpia pantalla
	Crsr	Cursor derecha
	Shift + crsr	Cursor izquierda
	Crsr	Cursor abajo
	Shift + crsr	Cursor arriba
	Ctrl + 9	Carácter inverso
	Ctrl + 0	Carácter normal
	Del	Borrar
	Shift + del	Insertar

Teclas de función

Cómo se ve	Cómo se teclea
	F1
	F2 = Shift + F1
	F3
	F4 = Shift + F3
	F5
	F6 = Shift + F5
	F7
	F8 = Shift + F7

commodore Magazine SERVICIO



Núm. 5
Programas, juegos y concurso/Londres: Quinta feria de Commodore/Basic, versión 4.75.



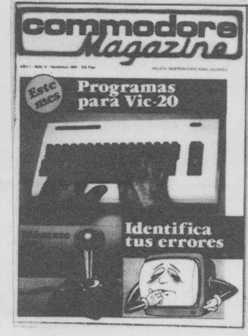
Núm. 6
El misterio del Basic/Lápices ópticos para todos/Concurso, juegos, aplicaciones.



Núm. 7
El ordenador virtuoso. MusiCalc. Programa monitor para el 64. Lápices ópticos. Ampliación de memoria para Vic-20.



Núm. 8
Joystick y Paddle para todos. Misterio del BASIC. EL LOGO. Cálculo financiero. Programas.



Núm. 9
Conversión de programas del Vic-20 al C-64. Móntale un paddle. Identifica tus errores. Software comentado.



Núm. 10
Koala Pad: La potencia de un paquete gráfico. Trucos. El FORTH. Software comentado. EL LOGO.



Núm. 11
Music-64. Supervivencia (1.ª parte). Cómo guarda el diskette la información. Sintetizador-64. El Forth (1.ª parte).

Núm. 12
Commodore-16 por dentro y por fuera. Sprites: los alegres duendecillos (1.ª parte). Supervivencia (1.ª parte). El Forth (y 3.ª parte).



Núm. 13
Análisis: programas de ajedrez. Los Cazafantasmas, 64. Vic en el espacio. La impresora que dibuja. Interface paralelo.



Núm. 14
Sprites; cómo entenderse con los duendes. Pilot: un lenguaje de alto nivel. Guía de Software para C-64.



Núm. 15
Síntesis de voz: su ordenador tiene la palabra. Pilot: un lenguaje de alto nivel (2.ª parte). Guía de software para C-64 (2.ª parte).



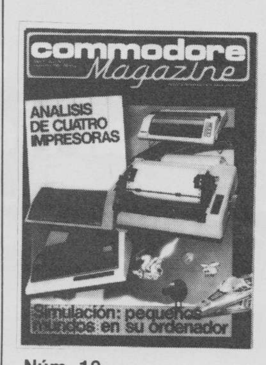
Núm. 16
Análisis de simuladores: vuela con tu C-64. Contabilidad para pequeños negocios. Cómo acelerar la ejecución de gráficos en BASIC. Submarino Commander. Pilot un lenguaje de alto nivel (3.ª parte).



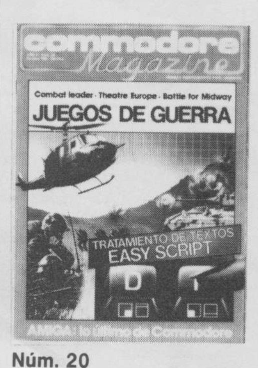
Núm. 17
Una lección de anatomía: los microordenadores por dentro. Bruce Lee: la furia oriental en el C-64. Quick Data Drive. Colossus Chess: un coloso del ajedrez.



Núm. 18
Practicalc: todo el poder de una hoja electrónica. Pascal (1.ª parte). Programas: juegos y aplicaciones veraniegos. ¡Canasta!: dos ases del baloncesto para el C-64.



Núm. 19
Análisis de cuatro impresoras. Simulación: pequeños mundos en su ordenador. Pascal (2.ª parte). Entombedy The Staff of Karnath: aventuras gráficas y mucha acción.



Núm. 20
Juegos de Guerra: Combat leader, Theatre Europe, Battle for Midway. Tratamiento de textos Easy Script. Amiga: lo último de Commodore. Libros, juegos y aplicaciones.



Núm. 21
Video-Digitizer: visión artificial para Commodore. Seikosha GP 700 VC: una impresora a todo color y con capacidades gráficas. Sprites multicolores. El nuevo C-128.

DE EJEMPLARES ATRASADOS

Complete su colección de COMMODORE MAGAZINE

A continuación le resumimos el contenido de los ejemplares aparecidos hasta ahora.



Núm. 22
Programas lightning: gráficos profesionales a tu alcance. Montaje: un interruptor programable para el C-64. Sprites múltiples. Cómo graba los datos el Datassette.



Núm. 25
Los ports: conexiones al exterior. El BASIC del C-128 Matemáticas por ordenador. Software educativo. Melodías musicales del Solfeo al Basic.



Núm. 27
Inteligencia Artificial: los lenguajes expertos. Sistemas de numeración: binario y hexadecimal. Comandos de disco C-128. Ficheros secuenciales. Libros, juegos y trucos.



Núm. 29
Piratas: problemas y legislación entorno a este tema. Backups: comentario del cartucho «Freeze Frame» y del Interface «Cosmos' Thoug». Introducción al PC de Commodore. Los juegos de Lucas Film.



Núm. 32
Gráficos en el Vic-20, C-64 y C-128. Hablemos de ordenadores o de cómo conseguir que no nos entienda nadie. Compatibles: fenómeno importante en el mundo de la microinformática profesional.



Núm. 23
Sinfonías en Chip: síntesis de sonido. Sprites en movimiento. Paisajes fractales en tu Commodore. Código máquina. Sight and sound: cuatro maestros de la música. Galería de Software: los mejores programas analizados en profundidad.



Núm. 26
Joysticks: ¿cómo son?, ¿cuáles son?, ¿cómo se programan?. Gráficos en código máquina. Simulador Spectrum. Proyecto Atenea.



Núm. 28
Robótica: evolución de la robótica y las posibilidades técnicas de los Robots. MECOMO: el brazo Robot. Mapa de memoria del C-128. Nuevas profesiones.



Núm. 31
La Magia del Amiga: primeras impresiones y posibilidades que este micro puede ofrecernos. Los periféricos que vienen. Mutaciones de personalidad.



Núm. 33
Reportaje fotográfico sobre la nueva imagen del C-64. Ofimática: la utilidad de los ordenadores en la empresa. Ordenadores de segunda mando. Juegos de Ingenio.



Núm. 24
El ordenador en la enseñanza. ADA para Commodore. El C-128 en fotos. Los peques y el ordenador. Seikosha Sp-1000 VC. Investigación sobre el sonido.

CORTE Y ENVIE ESTE CUPON A **COMMODORE MAGAZINE**

SERVICIO DE EJEMPLARES ATRASADOS

Bravo Murillo, 377. Tel.: 7337969 - 28020 MADRID

Ruego me envíen al precio de **300** ptas. los siguientes ejemplares atrasados de COMMODORE MAGAZINE __________

El importe lo abonaré

Contra reembolso ☐ Cheque adjunto ☐ Con mi tarjeta de crédito ☐ American Express

☐ Visa ☐ Interbank

Número de mi tarjeta ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

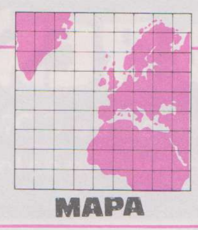
Fecha de caducidad __________

NOMBRE __________

DIRECCION __________

POBLACION __________ C.P. __________

PROVINCIA __________


MAPA


HABILIDAD


BELICO



FICHA DEL JUEGO

NOMBRE: «V»
DE: **ERBE**
ORDENADOR: **COMMODORE-64, 128**
CONTROL: **JOYSTICK**

PUNTUACION	1	2	3	4	5	6	7	8	9	10
ORIGINALIDAD							■			
ADICCION							■			
GRAFICOS									■	
SONIDO							■			
GENERAL								■		

Pues sí, esa «V» se refiere a aquella serie de televisión, que hace unos meses vimos en España. Todos los personajes de «V» aparecen en el juego. Han llegado a la Tierra los terribles reptiles, que camuflados bajo apariencia humana, intentan hacerse con nuestro planeta y esclavizar a sus habitantes.

Tu eres Donovan, el protagonista, y tu misión una vez dentro de la Nave Nodriza, es colocar explosivos en los puntos estratégicos e intentar volar esta fortaleza de los lagartos.

La perversa Diana ha ordenado a unos Robots que acaben contigo. Algunos te dispararán, pero todos emiten unas fuertes descargas eléctricas cuando los tocas, y cualquier contacto con ellos debilitará tu corazón. En la parte inferior de la pantalla un electrocardiograma te muestra la velocidad a la que late tu corazón; procura que sea lo más lenta posible, sin llegar a tener electrocardiograma plano, naturalmente.

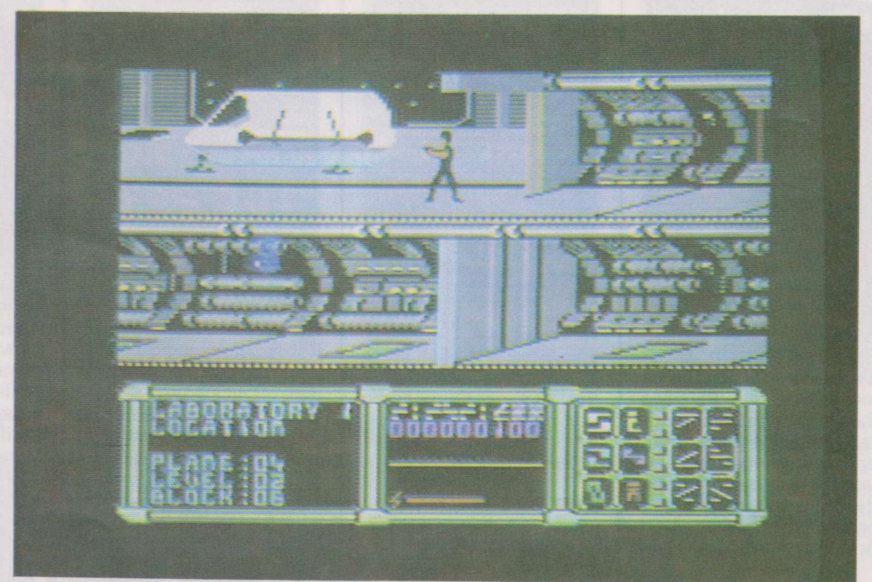
Además de este aparato, cuentas con otros indicadores que si utilizas bien, te mostrarán las diferentes zonas de la nave. Está formada por cinco planos verticales, a los que podrás acceder por las puertas laterales y utilizando algún código secreto que tendrás que descubrir.

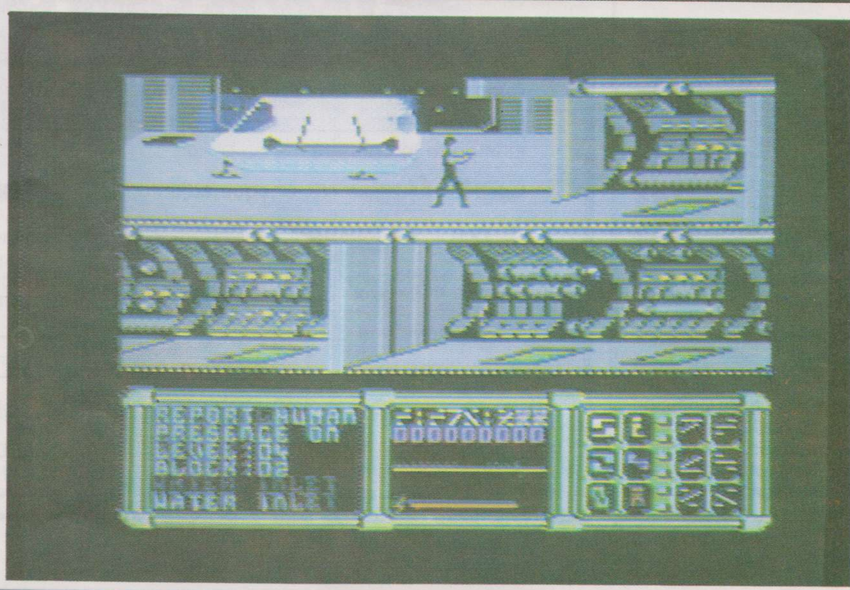
También dispones de una Zona de Texto, que te será útil en distintas ocasiones: colocar explosivos, obtener información, o tomar contacto con los ordenadores.

Pero si además de ir colocando los explosivos en el lugar acertado, vas buscando por los distintos laboratorios la fórmula del Polvo Rojo (que como sabes es un preparado mortal para los alienígenas), habrás conseguido librar al Planeta Tierra de estos horribles invasores.

Seguro que te sorprenderá la agilidad con que Donovan salta y se mueve a través de la Nave Nodriza, así como el gran número de posibilidades que encierra este juego.

La resistencia es fuerte. No descanses, vigila tu corazón, y ¡¡adelante, Donovan!!







Bomb-Jack



¿Quién es Bomb-Jack? en cuanto comience el juego lo reconocerás, pues no es otro que Super Ratón. ¿Te acuerdas?, es ese ratoncillo con una fuerza inusitada en los de su especie, que más de una vez ha ido en ayuda de sus congéneres más débiles, cuando se encuentran en apuros con los gatos.

En esta ocasión, los gatos, hartos ya de ser vencidos siempre por Super Ratón, han tomado medidas drásticas y han llenado el mundo de bombas. Nuestro amigo Bomb-Jack debe ser muy rápido para desactivarlas todas y así salvar el planeta.

Bomb-Jack comenzará su misión en las lejanas tierras de Egipto. Ante los ojos de la gran Esfinge tendrá que desconectar un total de veinticuatro bombas. Bomb-Jack puede moverse y volar en todas direcciones, así como dar un maravilloso salto que le permitirá alcanzar incluso las bombas más escondidas. Para que nuestro protagonista pueda efectuar este salto, tendrás que pulsar el botón de disparo de tu joystick y moverlo hacia arriba.

Pero, por desgracia, no se trata sólo de desactivar las bombas; hay peligros y dificultades mayores en todas y cada una de las pantallas que vayas atravesando.



Un extraño pajarraco no dejará de perseguirte por todo el mundo, y lo mismo harán unos robots, que al tomar contacto con el suelo se convierten en una especie de mariposas que pueden causarte graves daños. También deberás tener cuidado con las balas que «alguien» te envía, ya que si fueras alcanzado por una de ellas perderías una vida. Este es el precio que tendrás que pagar cada vez que tengas «un encuentro» con cualquier enemigo.

Durante la aventura debes estar bien atento a unas letras que encerradas en un círculo te manda una persona que quiere ayudarte. Cuando aparezca la letra P estás de suerte, puesto que mientras que esta letra permanezca en pantalla, tus enemigos no podrán atacarte, y puedes pasearte con toda libertad por Egipto, la gran ciudad, o cualquier otro lugar de! mundo en el que haya bombas, sin ser atacado.

Si la letra que aparece en el interior del círculo es una B, tampoco puedes quejarte, puesto que tu puntuación se multiplicará de dos a cuatro veces, mientras esta letra esté en pantalla.

Pero si eres tan afortunado que ves una letra E, a lo largo de tu recorrido, enhorabuena, habrás conseguido una vida extra.

Los gráficos te irán mostrando a lo largo del juego el lugar del mundo en que te encuentras. Están bastante bien conseguidos, pero al actuar como fondo de la acción, carecen de movimiento. Las figuras animadas son el protagonista y sus enemigos.

FICHA DEL JUEGO

NOMBRE:	**BOMB-JACK**
DE:	**ELITE** (Zafiro)
ORDENADOR:	**COMMODORE-64, 128**
CONTROL:	**JOYSTICK**

PUNTUACION	1	2	3	4	5	6	7	8	9	10
ORIGINALIDAD					■					
ADICCION								■		
GRAFICOS						■				
SONIDO							■			
GENERAL							■			





Hacker II

FICHA DEL JUEGO

NOMBRE:	**HACKER II**
DE:	**ACTIVISION (Proeinsa)**
ORDENADOR:	**COMMODORE-64, 128**
CONTROL:	**JOYSTICK O TECLADO (Imp. opc.)**

PUNTUACION	1	2	3	4	5	6	7	8	9	10
ORIGINALIDAD							■			
ADICCION								■		
GRAFICOS							■			
SONIDO						■				
GENERAL							■			

Seguro que aún recuerdas Hacker, ese juego de Activision en el que te veías envuelto en una complicada misión, al introducirte involuntariamente en un sistema de computadoras. Pues bien, parece que cumpliste bien tu cometido, puesto que has sido seleccionado para llevar a cabo otro importante trabajo.

Ha vuelto a ocurrir, esa extraña interferencia se ha introducido en tu ordenador... «LOGON PLEASE»... y parece que la cosa va muy en serio; el Gobierno de los Estados Unidos, te ha reconocido como una autoridad en sistemas de seguridad de computadoras, y solicitan tu ayuda para una misión de vital importancia para la C.I.A.

Si aceptas el reto, pulsa return en tu ordenador, si no, basta con desenchufarlo... de cualquier modo, la C.I.A. tiene ya datos sobre ti, y en cualquier momento podrás ser necesitado y tu ordenador será intervenido de nuevo.

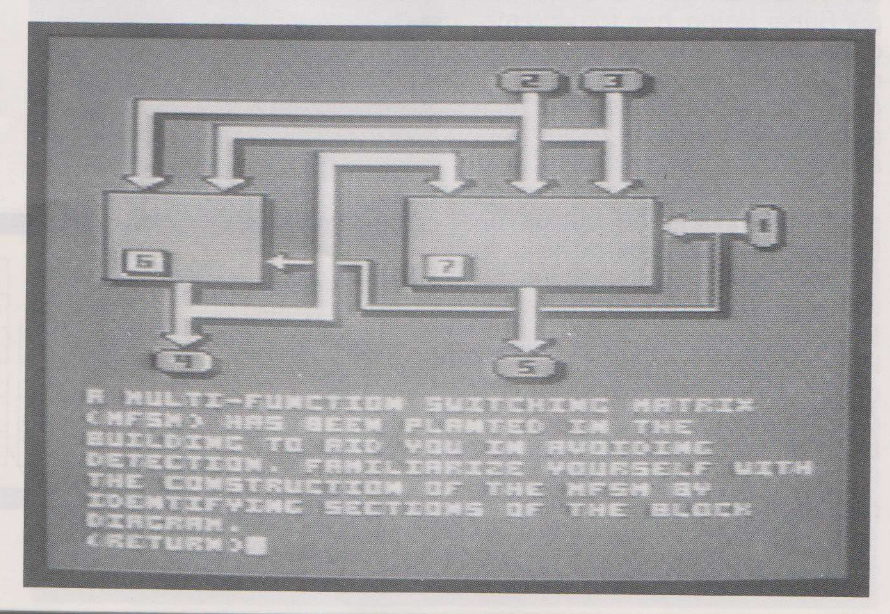
Pues bien, la misión es la siguiente: Alexandre Cherkazov, un científico ruso, especialista en política estratégica ha ideado un plan cuyo objetivo es derrocar al Gobierno de los Estados Unidos. Se cree que el plan está escondido en un documento llamado «Doonsday Paper», y la única copia se encuentra cuidadosamente guardada en una base militar rusa en Siberia. A primera vista, la misión parece casi imposible de realizar, pero con la ayuda de unos agentes de la C.I.A., que bajo la apariencia de guardias de seguridad se han infiltrado en la base rusa, tienes posibilidades de salir con éxito de tan complicado problema.

Estos agentes han sido capaces de localizar tres unidades de control remoto (MRU), dentro del complejo militar. Cada MRU ha sido equipado con un analizador óptico de control remoto, con el propósito de detectar a los intrusos. Cuentas con la información y ayuda que te proporcionan estos agentes, pero la sofisticación de los dispositivos anti-espías es altísima. Todas las precauciones son pocas.

Debes transportar una copia del documento etiquetado como «CLASSIFIED», desde el almacén hasta la entrada, donde te estará esperando un agente de la C.I.A. Las partes de la clave que te permitirán acceder al almacén se encuentran en cuatro cabinas a las que tendrás que llegar.

No sólo cuenta tu discreción, a la hora de enfrentarte con este trabajo, sino que será muy importante también tu capacidad para utilizar los dispositivos y cámaras de vídeo de la base militar, para ayudarte en tu búsqueda. Cuando te encuentres ante el panel de control, podrás disponer de cuatro pantallas, en las que verás gran parte del edificio, si consigues orientar bien las cámaras interiores y exteriores de control remoto. Además cuentas con un magnetoscopio, que te permitirá tener una visión de lo que ha sido captado por las distintas cámaras, incluso antes de que tú llegaras. La información en cintas de vídeo te está esperando, sólo es necesario que busques lo más importante en el momento oportuno.

Hacker II, no es un juego de sencillo manejo, y puede ser que te cueste un poco familiarizarte con el manejo de los aparatos que utilizarás en tu misión. Pero no dudamos que una vez superada esta primera fase puedas convertirte en un Hacker-adicto. □



SU PROGRAMA PARA CUALQUIER SISTEMA COMMODORE PUEDE HACERLE GANAR 5.000 PTAS.

EL PRESENTE CONCURSO ESTA ABIERTO A TODOS NUESTROS LECTORES Y SU PARTICIPACION E INSCRIPCION ES GRATUITA.
LEA LAS BASES DEL CONCURSO

- NO SE ESTABLECEN LIMITACIONES EN CUANTO A EXTENSION, TEMA ELEGIDO O MODELO DE ORDENADOR
- LOS CONCURSANTES DEBERAN ENVIARNOS A LA DIRECCION QUE FIGURA AL PIE, EL CASSETTE O DISKETTE CONTENIENDO EL PROGRAMA, UNA EXPLICACION DEL MISMO Y, AL SER POSIBLE, UN LISTADO EN PAPEL DE IMPRESORA. SE PODRAN ENVIAR TANTOS PROGRAMAS COMO SE DESEE
- LOS PROGRAMAS, PREVIA SELECCION, SERAN PUBLICADOS EN LA REVISTA, OBTENIENDO TODOS ELLOS 5.000 PTAS.
- LA DECISION SOBRE LA PUBLICACION O NO DE UN PROGRAMA CORRESPONDE UNICAMENTE AL JURADO NOMBRADO AL EFECTO POR "COMMODORE MAGAZINE", SIENDO SU FALLO INAPELABLE
- LOS CRITERIOS DE SELECCION SE BASARAN EN LA CREATIVIDAD DEL TEMA ELEGIDO Y LA ORIGINALIDAD Y/O SENCILLEZ EN EL METODO DE PROGRAMACION GLOBAL
- ENVIAR A:
CONCURSO COMMODORE MAGAZINE



commodore Magazine
C/BRAVO MURILLO, 377 5.° A 28020 MADRID

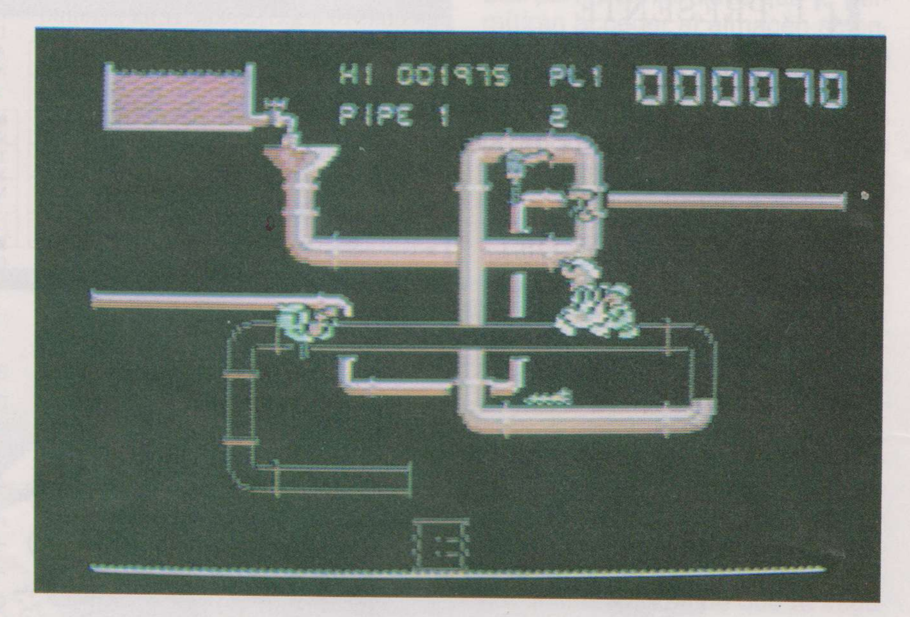

AZAR


HABILIDAD

Pipeline

Cuando te hayas cansado de hacer viajes espaciales, de perseguir y ser perseguido por los más extraños seres, de recorrer peligrosos laberintos, etc., entonces puedes dedicarte, por ejemplo, a arreglar tuberías. No, no estamos tratando de tomarte el pelo, es que para llevar a cabo la misión que ahora nos propone Mastertronic tendrás que convertirte en fontanero.

Tu misión en Pipeline es llenar de agua unos cuantos bidones, que se mueven por una cinta de transporte en el suelo del edificio. Para ello tendrás que ir abriendo las tuberías a través de varias pantallas.



Naturalmente, te encontrarás con algunas pequeñas dificultades que complicarán tu trabajo. Las herramientas, cansadas ya de obedecer siempre, han decidido trabajar por su cuenta, y en ocasiones, estropean la labor que tú con tanto esfuerzo has conseguido llevar a cabo. Un auténtico motín de taladradoras, martillos y sopletes se ha desencadenado, y debes defenderte de ellos disparando sin piedad.

Las teclas de función te permiten optar por la modalidad de uno o dos jugadores, cambiar tu número de vidas o hacer una pausa para irte a merendar (tu amigo el fontanero te esperará.)

Cuentas, por otra parte, con el apoyo de dos fieles ayudantes que te seguirán por donde quiera que vayas. Estos chinitos trabajadores podrán reparar los desperfectos causados por las máquinas, y se pondrán contentísimos cada vez que consigas llenar los bidones de una pantalla.

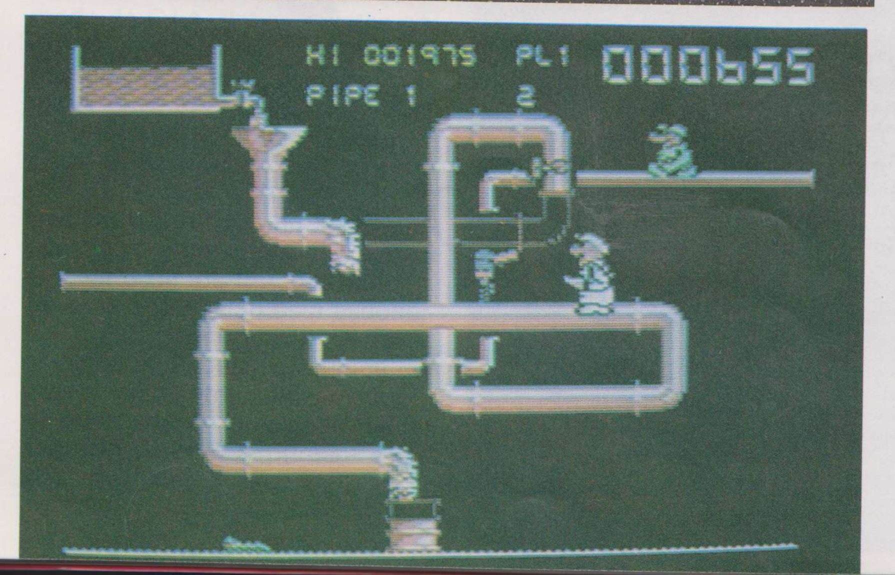
A medida que avanzas en el juego, irán apareciendo distintos ejemplos, y la tubería será más y más larga. Con un poco de entrenamiento en las primeras pantallas, no te será difícil ser un buen fontanero.

Pipeline es un juego sencillo y sin pretensiones de espectacularidad, tanto en sus gráficos y sonidos como en su argumento. Pero ¡ojo!, es también un juego que puede crear alto grado de adicción.

FICHA DEL JUEGO

NOMBRE:	PIPELINE
DE:	MASTERTRONIC (Drosoft)
ORDENADOR:	COMMODORE-64, 128
CONTROL:	JOYSTICK Y TECLADO

PUNTUACION	1	2	3	4	5	6	7	8	9	10
ORIGINALIDAD								■		
ADICCION									■	
GRAFICOS							■			
SONIDO						■				
GENERAL								■		




INGENIO


AZAR


AVENTURAS

Zzzz

Mmm... hoy sí que estoy cansado... vaya un día... realmente necesito dormir, aah... por fin puedo descansar sobre la almohada... qué sueño... zzz... zzz

Oh!, qué sed más horrible, me levantaré a por un vaso de agua, con el sueño que tengo... Qué es eso!! estoy en medio del Oceano! Socorro!! ¿Cómo he venido a parar aquí? Bueno eso no importa ahora, lo que debo hacer es buscar el modo de salir.

Han aparecido unos extraños símbolos, ¿qué querrán decir? vamos a ver, esas cuatro flechas indican la dirección en que puedo moverme, y esos dos cuadrados los podré utilizar para entrar o salir de algún sitio... puedo examinar objetos, grabar, ver, coger o dejar cosas.

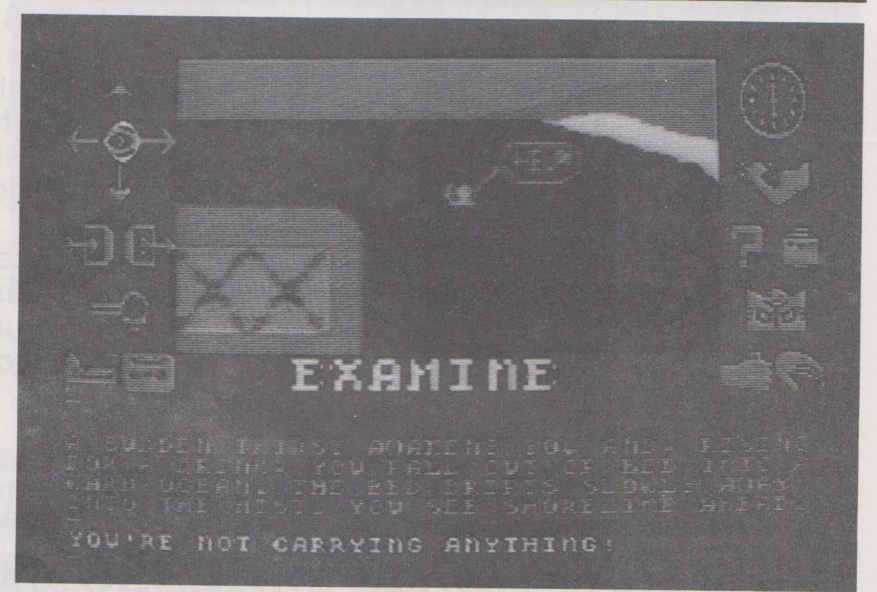
Iré hacia el Norte... hum... extraños paisajes, objetos, tengo muchas opciones ¿cogeré esa herramienta? quizás sea útil, no sé... oh!! qué he hecho... ahora estaré perdido en el mundo de los sueños para el resto de mi vida!!!...

Si no quieres que te ocurra lo mismo que a este inexperto jugador, procura leer bien las instrucciones, y estar muy atento cuando juegues con Zzzz. Te moverás por el mundo de los sueños, las cosas, en un principio, no parecen tener mucho sentido, pero pueden adquirirlo si eres observador.

FICHA DEL JUEGO

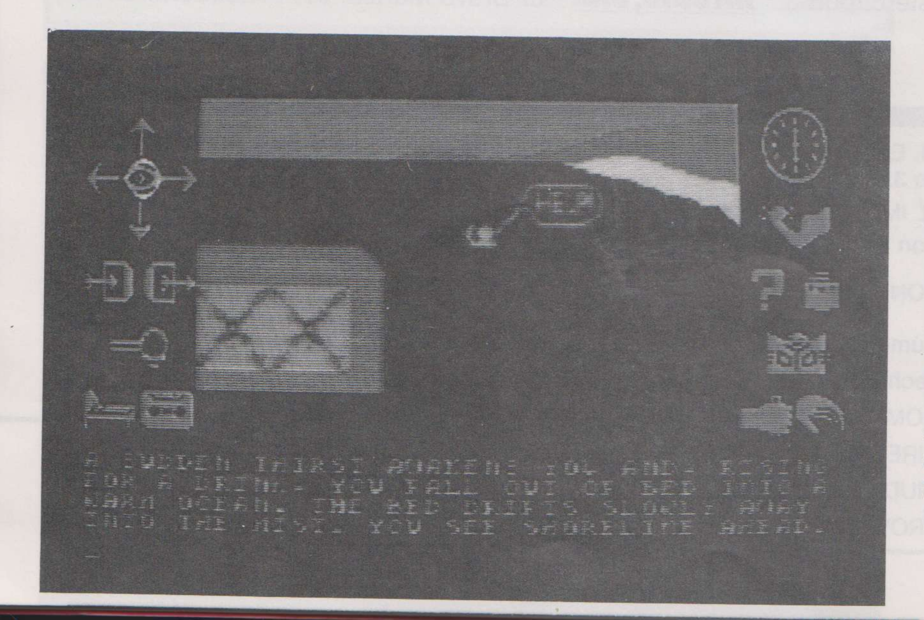
NOMBRE: Zzzz
DE: MASTERTRONIC (Drosoft)
ORDENADOR: COMMODORE-64, 128
CONTROL: JOYSTICK O TECLADO

PUNTUACION	1	2	3	4	5	6	7	8	9	10
ORIGINALIDAD									■	
ADICCION								■		
GRAFICOS								■		
SONIDO						■				
GENERAL								■		



Los creadores de Mastertronic han dejado correr sin miedo su imaginación (tan calenturienta en ocasiones) y seguro que conseguirán sorprenderte en tu recorrido por este extraño lugar. Puedes tomar muchas opciones. Procura ir recordando lo que ocurre cada vez que tomas un camino diferente. Cuando hayas avanzado mucho, puedes grabar estos datos en la cinta del juego y continuar por ese camino la próxima vez que lo intentes. Si lo haces así ahorrarás mucho tiempo y conseguirás avanzar sobre tus sueños por el camino acertado.

Te esperan muchas pantallas y sorpresas en Zzzz, no desesperes al principio y conseguirás dominar el mundo de los sueños. ¡Suerte!





Street Surfer

Hemos visto juegos de carreras de coches, motos, con distintos circuitos... pero hasta ahora no había llegado a nuestras manos un juego en el que hay que intentar batir el record de velocidad y permanencia en... un monopatín.

Te paseas con tu monopatín por una transitada carretera, así que no sólo tendrás que mantener el equilibrio y correr todo lo posible, también deberás ser ágil para no ser atropellado por los coches y camiones que, como tú, circulan por la calzada.

En la parte inferior de la pantalla verás unos mensajes mediante los cuales podrás informarte de tu situación y posibilidades. En caso de que un automóvil se encuentre a tu espalda, serás avisado de que corres el riesgo de ser atropellado.

Si además de ser un experto en el manejo del monopatín, te comportas como un buen ciudadano, recoges las botellas vacías que te encuentres en tu camino

Algunos conductores, conscientes del esfuerzo que estás realizando, te ofrecerán bebidas al pasar. Si bebes resistirás más y también acumularás un mayor número de puntos en el marcador.

Los expertos en este arte del monopatín aconsejan no ir delante de ningún vehículo, ni atravesar las manchas de la carretera, procura esquivarlas o saltarlas, de lo contrario resbalarás e inevitablemente caerás al suelo.

El trepidante sonido que te acompaña en tu recorrido proporciona una emoción adicional al juego. Según vayas aumentando tu velocidad, los riesgos son mayores. No te dejes llevar por la tensión y maniobra cuidadosamente.

FICHA DEL JUEGO

NOMBRE:	**STREET SURFER**
DE:	**MASTERTRONIC (Drosoft)**
ORDENADOR:	**COMMODORE-64, 128**
CONTROL:	**JOYSTICK**

PUNTUACION	1	2	3	4	5	6	7	8	9	10
ORIGINALIDAD						■				
ADICCION								■		
GRAFICOS							■			
SONIDO							■			
GENERAL							■			

Las cuatro posiciones del *joystick* te permitirán moverte en distintas direcciones, y el botón de disparo una vez pulsado, te hará dar un gran salto. Sólo una advertencia, puedes no acelerar, pero nunca retroceder, ya que la posición del *joystick* hacia atrás te hará flexionar las rodillas y adoptar una postura más apropiada para correr en tu monopatín.





David Bowie me miraba fijamente. El, el rey de los Ewoks, me juraba que poseía un castillo tan difícil de encontrar que jamás nadie podría llegar a él, si él no lo permitía. El palacio estaba rodeado por un inmenso y complicado laberinto que no había persona alguna que pudiera dar con la solución.
—Curioso reto —pensé yo.

Por supuesto y por fortuna, la amenaza no iba dirigida contra mí, sino contra una bella princesita a la que el estrambótico monarca había robado un niño. A pesar de todo, tomé la causa como mía y a modo de informático caballero andante comencé a pensar sobre cómo es factible resolver un laberinto por ordenador.

Supongamos un caso real: «las antiguas tumbas egipcias». Los faraones, temerosos de que los ladrones pudieran robarles las riquezas con las que eran enterrados, solían construir una enorme pirámide que contuviera su tumba y en la que casi más del 80 por ciento del espacio lo constituían oscuros y cavernosos pasadizos que ocultaban los tesoros a los posibles amantes de la propiedad ajena.

Entremos en una de ellas, hace dos mil años.

Está oscuro, así que usaremos una linterna.

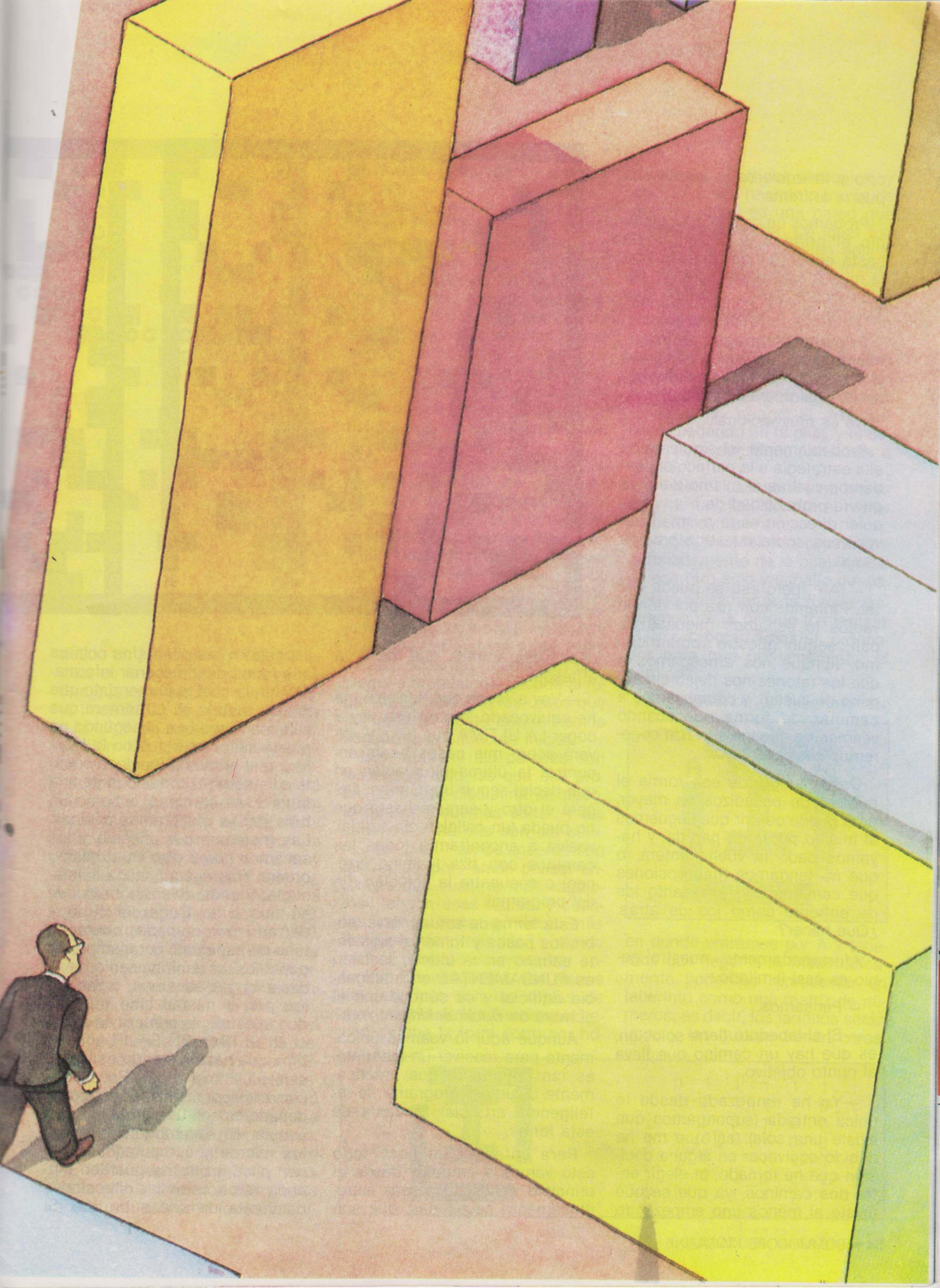
No, claro; hace dos mil años no se usaban esas cosas, así que optaremos por una vieja lámpara de aceite, pero sin genio dentro.

Encendemos y ¿qué hay?

Evidentemente, pasadizos. Probablemente, para empezar dispondremos de un surtido de caminos variados, uno a la derecha,



Dentro del laberinto



otro a la izquierda, y quizás una puerta al frente.

Probamos un camino, por ejemplo el primero, y proseguimos hasta encontrarnos otra vez con la misma alternativa, otra vez decidir sobre varios caminos que salen frente a nosotros.

Primera estrategia: el movimiento aleatorio, o la «intuición», y ante algo que hay que decidir, tomar cualquier camino.

No es muy racional.

Probablemente, si siguiéramos esa estrategia a la perfección, daríamos vueltas al mismo sitio, ya que la probabilidad de ir en cualquier dirección es la misma. Volveríamos sobre nuestros propios pasos.

—¡Ah!, pero eso se puede evitar. Pintamos con tiza por donde vamos (o echamos miguitas de pan, según nuestro romanticismo, aunque nos arriesgamos a que los ratones nos dejen sin camino de vuelta), y comenzamos a caminar, de forma que cuando veamos un camino con tiza cogeremos cualquier otro.

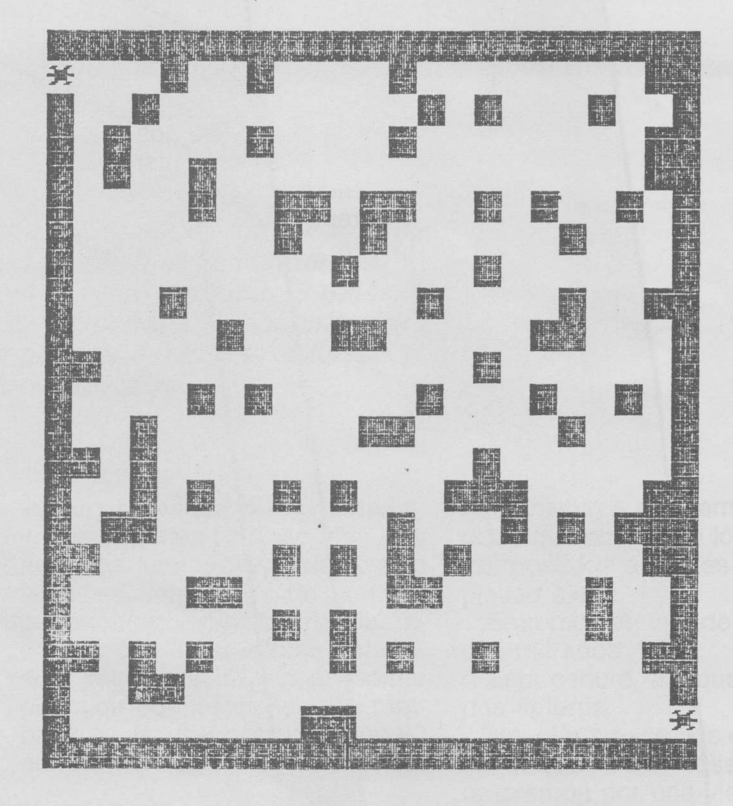
Desde luego, de esa forma el número de pasadizos es mayor, pero puede ocurrir que lleguemos al mismo punto de partida, y hayamos dado la vuelta entera, o que no tengamos más opciones que caminos con tiza, tanto los de enfrente como los de atrás. ¿Qué hacer?

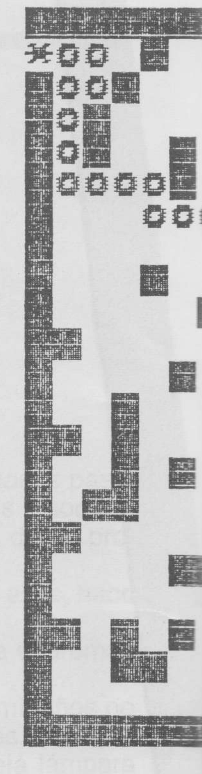
Afortunadamente, nuestro genio es casi ilimitado.

—Pensamos.

—Si el laberinto tiene solución, es que hay un camino que lleva al punto objetivo.

—Yo he empezado desde la única entrada (supongamos que existe una sola), así que me he debido equivocar en alguna decisión que he tomado, al elegir entre dos caminos, ya que sé que existe al menos uno empezando





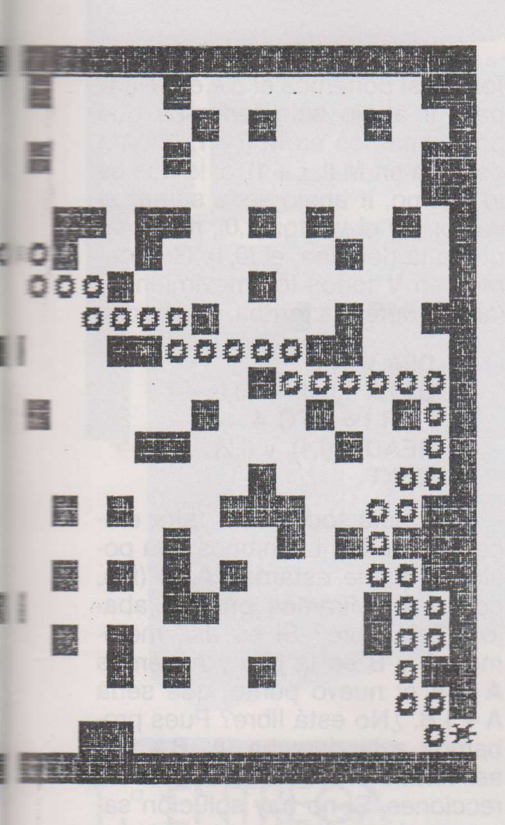
desde esa entrada que me lleva al tesoro.

—Voy a ver en qué decisión me he equivocado, y para eso voy a coger LA ULTIMA que hice, y volveré sobre mis pasos hasta encontrar la última bifurcación en que decidí seguir un camino. Cogeré el otro y seguiré hasta que no pueda (un callejón sin salida), vuelva a encontrarme todos los caminos con tiza (camino erróneo) o encuentre la solución (fin del programa).

Esta forma de actuar, volver sobre los pasos y tomar el siguiente camino en la última decisión es FUNDAMENTAL en inteligencia artificial y se conoce con el nombre de *Backtracking.*

Aunque aquí lo veamos únicamente para resolver un laberinto, es tan importante que prácticamente cualquier programa de inteligencia artificial funciona de esta forma.

Para entender un poco todo esto vamos a empezar desde el principio viendo dos cosas extremadamente necesarias, que son las colas y las pilas. Una cola es una forma de almacenar información en la cual el primer dato que hemos metido es el primero que sale, de forma que el segundo no puede salir hasta que no lo haya hecho el primero. Funciona como la cola de un autobús o la de una taquilla de entradas, o como un tubo donde metiéramos canicas. Las metemos por un sitio y las sacamos por el otro en el mismo orden. Hasta aquí, nada interesante. Vamos a ver ahora lo que es una pila. Coger el tubo y cerrarlo por un lado, como un tubo de aspirinas, por ejemplo. Si ponemos las aspirinas en orden y después las sacamos, como salen por el mismo sitio que por donde entran, la primera en entrar es la ULTIMA EN SALIR, y la última que habéis metido es la que sale en primer lugar. Pues bien, este concepto es usado en los ordenadores con una profusión que asusta. El Commodore, y todos los micros y computadores, tienen pilas en donde guardan muchos datos, como las direcciones de vuelta de una subrutina. Si



desde un programa BASIC llamamos en la línea 100 a una subrutina que empieza, por ejemplo, en la línea 1.700, el ordenador salta a la 1.700 y guarda en la pila la dirección 100, de forma que cuando la subrutina acabe y el ordenador encuentre RETURN, sacará de la pila la última dirección (en este caso la 100) y saltará hacia ella. ¿Y por qué una pila y no una cola, o un solo *byte*? pues muy simple. Si, en el ejemplo anterior, estando ejecutando la subrutina que empieza a partir de la 1.700, hay una línea, pongamos la 1.760, que llama a otra subrutina (por ejemplo, a partir de la 2.500), el ordenador tiene que guardar esa dirección (la 1.760) en la pila, encima de la anterior (la 100), y así cuando llegue al final de la segunda subrutina (el primer RETURN a partir de la 2.500) el ordenador sacará la última dirección que metió (la 1.760), proseguirá a partir de la siguiente (por ejemplo, la 1.770), y al encontrar de nuevo RETURN ejecuta lo mismo, saca de la pila la dirección 100 y salta a ella. Así es como todos los ordenadores trabajan con las subrutinas.

¿Y eso para qué nos vale a nosotros? Pues está claro. Cuando nos metamos en la pirámide nos llevaremos una «pila», en donde iremos guardando los lugares en donde hemos tomado una decisión, y al llegar a un punto donde no podamos continuar, «sacaremos» de la pila el último lugar donde decidimos y cogeremos otra dirección. Si esta segunda fuera un callejón sin salida, volveríamos a sacar de la pila para ver la anterior, etc.

Esto no sólo se usa para un laberinto. Multitud de procesos mentales en la resolución de problemas conllevan esta actuación, o al menos son simulables con ella. Cualquier problema se puede descomponer en muchos caminos posibles, donde algunos llevan a la solución, que es el objetivo buscado, y otros no, donde debemos dar marcha atrás. Realmente, podemos igualar un problema a un laberinto, donde el lugar de entrada son los datos que conocemos, hasta donde hemos llegado, o la situación inicial.

Los muros y las paredes son las reglas bajo las que ese problema o esa disciplina se rigen, lo que no podemos saltar, y los caminos entre ella son las soluciones que están de acuerdo con las reglas (podíamos pensar entonces que cuantos más «agujeros» tengan esas reglas, y cuantas menos halla, es decir, más flexibles sean, más fácil y rápido será el proceso de encontrar solución. En el ajedrez, los muros del laberinto son las reglas de movimiento de las piezas, y el hecho de que la torre corra por horizontales y no por verticales, y que no pueda hacerlo a través de una diagonal, es una pared que no podemos saltar. De igual forma, la pila sería también aplicable al juego, ya que para encontrar la solución (matar al rey) probamos caminos, y metemos y sacamos en la pila. Veamos un ejemplo:

— Tiene el rey desprotegido ese flanco. Moveré mi torre en ese lugar (metemos en la pila).

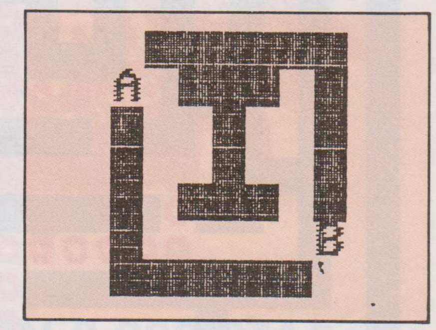
— Luego pongo el alfil ahí (sigo metiendo en la pila), y el caballo para dar jaque (uno más a la pila).

— No, el caballo ahí no (sacamos de la pila), porque él me le comería con la dama.

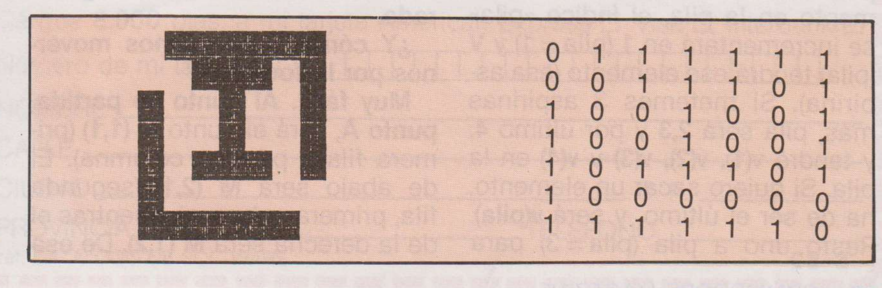
— No, no. Mejor (sacamos embarulladamente de la pila), el alfil en ese otro sitio y caballo detrás.

Pero volvamos a la tumba (saquemos de la pila de los temas). ¿Cómo programamos un ordenador para que nos encuentre el camino?

Supondremos un laberinto así:

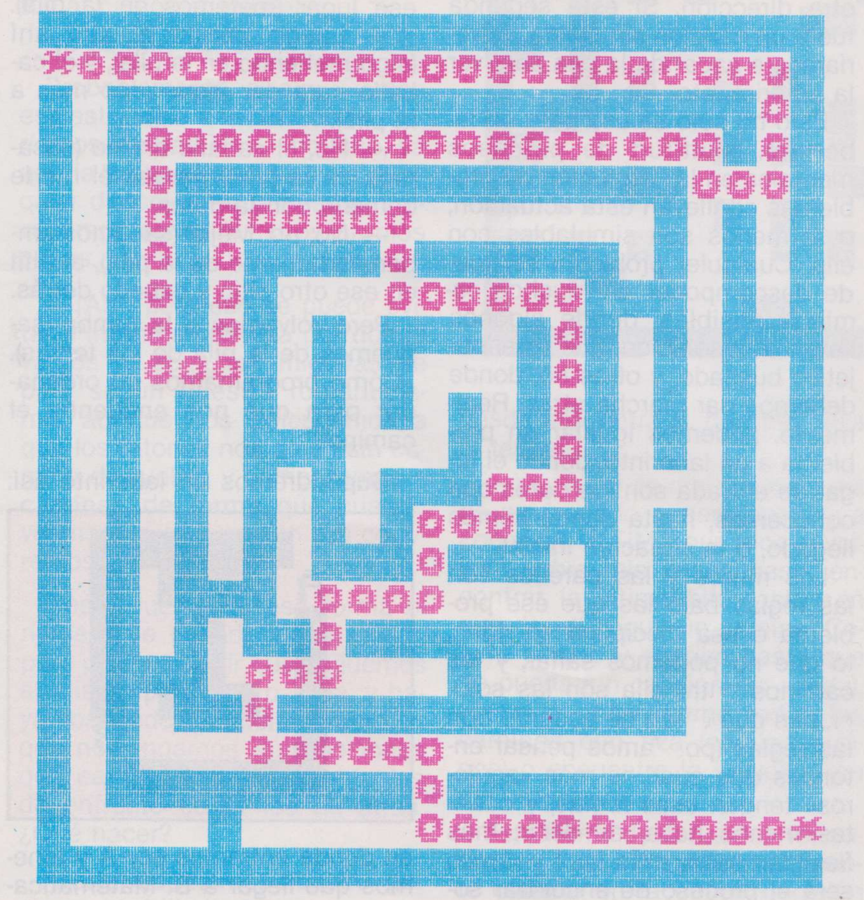


en donde entramos por A y tenemos que llegar a B. Matemáticamente podemos representar un laberinto como una matriz de números, es decir, los ladrillos serán unos y los huecos serían ceros.



0	1	1	1	1	1	1
0	0	1	1	1	0	1
1	0	0	1	0	0	1
1	0	0	1	0	0	1
1	0	1	1	1	0	1
1	0	0	0	0	0	0
1	1	1	1	1	1	0

Viajaremos entonces entre los ceros. Cada número, cada «paso», lo consideraremos un lugar en el que hemos decidido, ya que en cada uno de ellos podemos ir abajo, a la derecha, arriba y a la izquierda, así que a medida que avanzamos un paso, lo metemos en la pila. Esta la podemos simular con un vector **V**, y la tapa de la pila, es decir, la última aspirina del tubo, la guardaremos



con un número que apunta a ésta. Por ejemplo, si llamo a ese número como pila, y está vacía, pila será **0**. Si metemos un elemento en la pila, el índice «pila» se incrementará en **1** (pila = 1) y **V** (pila) tendrá ese elemento (esa aspirina). Si metemos 3 aspirinas más, pila será **2,3** y por último **4**, y tendré **v(1), v(2), v(3)** y **v(4)** en la pila. Si quiero sacar un elemento, ha de ser el último, y será **v(pila)**. Resto uno a pila (pila = 3) para que siga apuntando al último (el tercero) la próxima vez que extraiga un elemento. De esta forma tan simple hemos construido una pila donde guardaremos los puntos por donde hemos pasado. Más fácil aún es generar nuestro propio laberinto:

— FOR I = 1 TO TAMAÑO
— FOR J = 1 TO TAMAÑO
— M (I, J) = 0
— IF RND (0) 〉 0.8 THEN M (I, J) = 1
— NEXT J
— NEXT

Ya tenemos el laberinto generado.

¿Y cómo conseguimos movernos por la fortaleza?

Muy fácil. Al punto de partida, punto A, será el punto **M (1,1)** (primera fila y primera columna). El de abajo será **M (2,1)** (segunda fila, primera columna), mientras el de la derecha será **M (1,2).** De esa forma, si ponemos el pie en **M (i,j)**, para ir abajo sólo tenemos que posicionarnos en **M (i + 1, j)**, y la derecha en **M (i, j, + 1)**, o, lo que es lo mismo, ir abajo sería sumar al vector (i,j) el vector (1,0), mientras que a la derecha, el (0,1). Guardamos en **V** todos los movimientos (abajo, derecha, arriba, izquierda):

— DIM V (4,2)
— DATA 1,0,0,—1,0,0,—1
— FOR I = 1 TO 4
— READ v (I,1), v (I,2)
— NEXT

Y ya está todo. Sólo falta empezar a andar. Llamamos a la posición donde estamos **A, B** (fila, columna). Miramos primero abajo. ¿Está libre? Si es así, metemos **A** y **B** en la pila y hacemos **A** y **B** el nuevo punto, que sería **A + 1,B.** ¿No está libre? Pues probamos a la derecha (**A, B + 1**) y así vamos probando todas las direcciones. Si no hay solución sacamos de la pila el punto anterior y vemos en otra dirección. En el listado del programa, aparte de guardar el punto anterior, también guardamos cual fue la última dirección a la que miramos, para no tener que repetir. Si en el punto anterior miramos abajo y derecha, pues aparte de guardar la posición del punto, también guardamos que la próxima vez hay que volver, qué ocurrirá si el camino que tomamos no era el correcto y tuvimos que volver sobre nuestros pasos, nos tocará mirar arriba.

¿Qué fácil? —Pensé.

Pero... y si el punto que queremos encontrar está en el centro del laberinto... ¿Pasaría lo mismo?

Pues sí. Lo único que ocurre es...

—¡Eh, eh! que se ha acabado la película... ¿Te ha gustado?

—Sí, sí, claro... Por supuesto.. Muy curioso esto de los laberintos mentales y las tumbas egipcias...

—¿Qué? ¿Qué tumbas...? ¡Tú si que tienes laberintos mentales...!

Catálogo de Software para ordenadores personales IBM



Todo el Software disponible en el mercado reunido en un catálogo de 800 fichas

1.ª ENTREGA
550 FICHAS
+ FICHERO

Resto en dos entregas trimestrales de 150 fichas cada una



PRECIO TOTAL DE LA SUSCRIPCION 8.000 PTAS.

COPIE O RECORTE ESTE CUPON DE PEDIDO

CUPON DE PEDIDO

SOLICITE HOY MISMO EL CATALOGO DE SOFTWARE A:

infodis, s.a.

Bravo Murillo, 377, 5.º A
28020 MADRID

O EN CONCESIONARIOS IBM

El importe lo abonaré POR CHEQUE ☐ CONTRA REEMBOLSO ☐ CON MI TARJETA DE CREDITO ☐

Cargue **8.000** ptas. a mi tarjeta American Express ☐ Visa ☐ Interbank ☐

Número de mi tarjeta ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

NOMBRE ____________________

CALLE ____________________

CIUDAD ____________________ C. P. __________

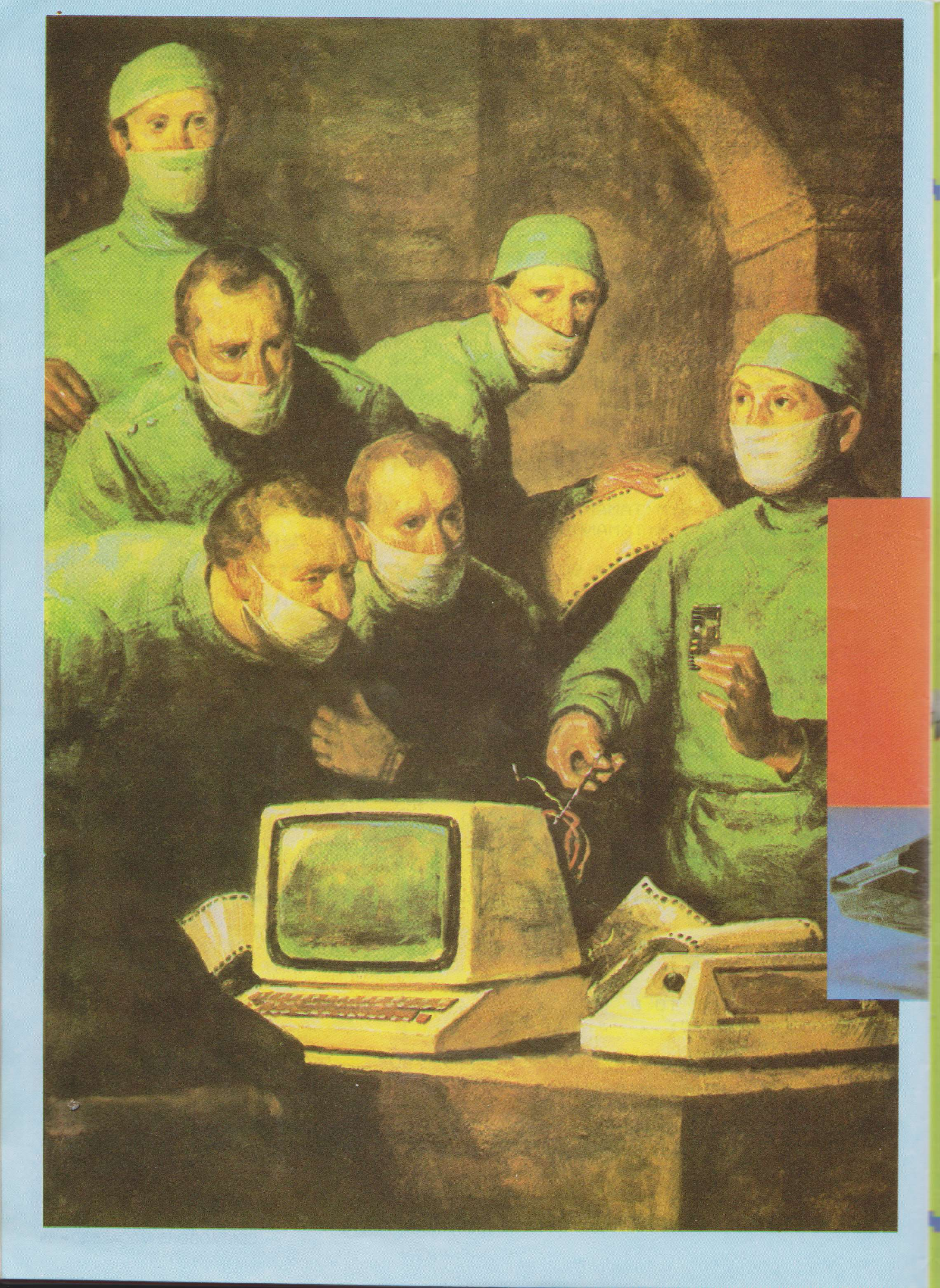
PROVINCIA ____________________ TELEFONO __________

ref: CATALOGO DE SOFTWARE

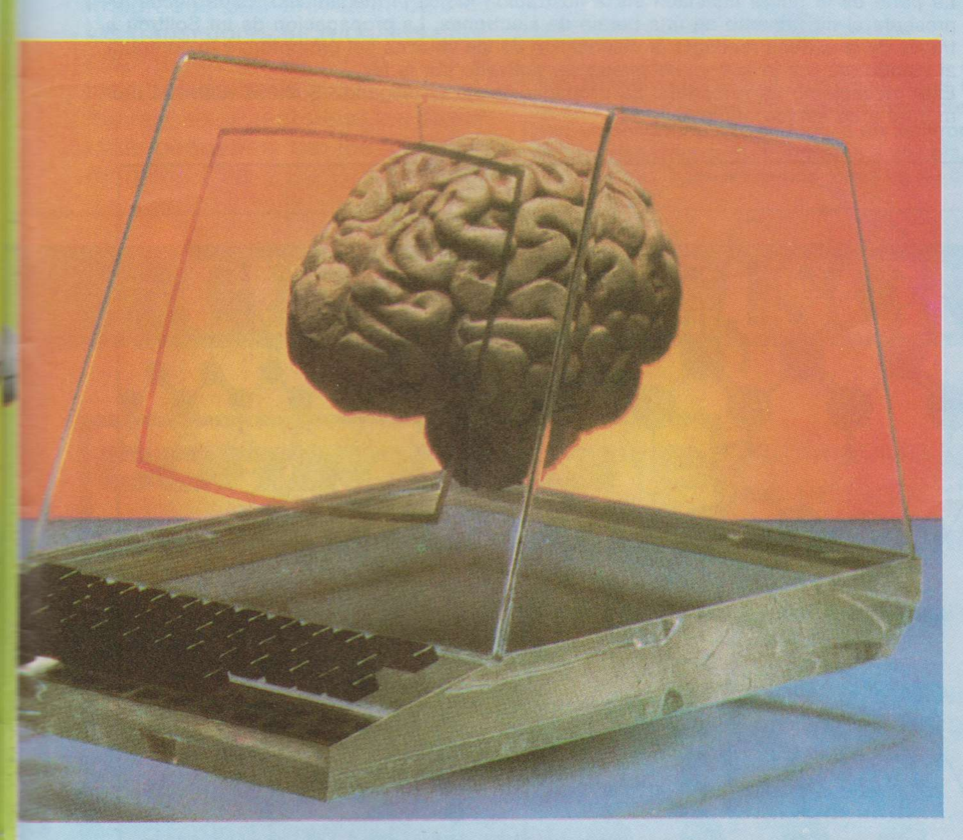
CS-2

```
5 POKE 53280,6 :PRINT"♥"
10 INPUT" QUIEN HACE EL LABERINTO, TU O YO ";Z$
20 INPUT" TAMANNO DE 4 A 23 ";TAMANNO
30 DIM P(300,3)
40 DIM M(TAMANNO+1,TAMANNO+1)
50 DIM V(4,2)
60 DATA 0,1,1,0,0,-1,-1,0
70 FOR I=1 TO 4 :READ V(I,1),V(I,2) :NEXT
80 IF Z$="YO" THEN GOSUB 2000
85 IF Z$<>"YO" THEN GOSUB 1000
90 GOSUB 3000
95 A=2 :B=2 :N=1 :PILA=1
100 FOR K=N TO 4
110 IF K>4 THEN GOTO 150
115 C=A+V(K,1)
120 D=B+V(K,2)
130 IF M(C,D)=0 THEN GOTO 500
140 NEXT K
150 PILA=PILA-1
160 IF PILA=0 THEN PRINT"EL LABERINTO NO TIENE SOLUCION":END
170 M(A,B)=9
180 A=P(PILA,1)
190 B=P(PILA,2)
200 N=P(PILA,3)+1
210 GOTO 100
500 P(PILA,1)=A
510 P(PILA,2)=B
520 P(PILA,3)=K
530 PILA=PILA+1
535 M(A,B)=K+2
540 IF C=TAMANNO-1 AND D=TAMANNO-1 THEN GOSUB 4000
550 A=C
560 B=D
570 N=1
580 GOTO 100
999 REM *********** LABERINTO AZAR ***********
1000 FOR I=1 TO TAMANNO
1010 FOR J=1 TO TAMANNO
1020 M(I,J)=0
1030 IF RND(0) > 0.8 THEN M(I,J)=1
1040 NEXT J
1050 NEXT I
1060 FOR I=1 TO TAMANNO
1061 M(I,1)=1 :M(1,I)=1 :M(TAMANNO,I)=1 :M(I,TAMANNO)=1
1062 NEXT I
1070 M(2,2)=0 :M(TAMANNO-1,TAMANNO-1)=0
1080 M(2,1)=9 :M(TAMANNO-1,TAMANNO)=9
1090 RETURN
```

```
1999 REM *********** LABERINTO MANUAL ***********
2000 CO=0
2010 PRINT :PRINT" TIENES QUE RELLENAR EL LABERINTO"
2011 PRINT" UTILIZANDO 0=VACIO Y 1=MURO ":PRINT
2020 FOR I=2 TO TAMANNO-1 :CO=CO+1
2021 PRINT CO;"FILA DE";TAMANNO-2;"CARACTERES";
2022 INPUT A$
2030 FOR J=2 TO TAMANNO-1
2040 B$=MID$(A$,J,1)
2050 M(I,J)=VAL(B$)
2055 NEXTJ,I
2060 FOR I=1 TO TAMANNO
2061 M(I,1)=1:M(1,I)=1:M(TAMANNO,I)=1:M(I,TAMANNO)=1
2062 NEXT I
2070 M(2,2)=0:M(TAMANNO-1,TAMANNO-1)=0
2080 M(2,1)=9:M(TAMANNO-1,TAMANNO)=9
2090 RETURN
2999 REM *********** SUBRUTINA GRAFICA ***********
3000 PRINT"▛" :FOR I=1 TO TAMANNO
3010 FOR J=1 TO TAMANNO
3020 IF M(I,J)=0 THEN PRINT " ";
3021 IF M(I,J)=9 THEN PRINT "▀*▛";
3025 IF M(I,J)<>0 AND M(I,J)<>9 THEN PRINT"█▜ ▀▛";
3030 NEXT J:PRINT"█"
3040 NEXT I
3050 RETURN
3999 REM *********** PINTA RECORRIDO ***********
4000 PRINT"█"
4010 FOR I=1 TO 300
4020 IF P(I,1)=0 THEN GOTO 6000
4030 A=P(I,1):B=P(I,2)
4040 GOSUB 5000
4050 NEXT
4999 REM **************** PINTA BOLA *****************
5000 PRINT"█"
5010 FOR N=2 TO A
5020 PRINT"█";
5030 NEXT
5040 FOR N=2 TO B
5050 PRINT"█";
5060 NEXT
5070 PRINT"▀O▛"
5080 RETURN
5999 REM ****************** FINAL *******************
6000 A=TAMANNO-1:B=TAMANNO-1
6010 GOSUB 5000
6020 END
```



Los biochips



Algún día, las moléculas orgánicas podrían reemplazar al silicio en los chips que gobiernan el funcionamiento de los ordenadores. Esta audaz hipótesis tiene poco que ver con la ciencia-ficción y mucho con los vertiginosos avances de la ingeniería genética. El tema, no sólo tecnológico, plantea dilemas éticos de envergadura, como éste expuesto por James McAlear, uno de los especialistas que trabajan en los futuros biochips «Nuestra civilización posee el potencial para crear una forma de vida muy superior.»

La idea de crear un ordenador biológico que pudiera emular el cerebro humano ha estado vigente desde la década de los 50, cuando los ordenadores se hicieron realidad. Vivimos en un mundo que no ve ningún fin al logro científico, y seguramente, como dicen los escritores de ciencia-ficción, sólo el tiempo resiste ante este hecho.

Al menos un científico estaría de acuerdo con esta determinación. «Verdaderamente, no estamos tratando con la ciencia-ficción en nuestra investigación», comenta el doctor **James McAlear,** presidente y jefe del equipo científico de **Gentronix Laboratories,** una pequeña firma de ingeniería genética en Rockville, Maryland. «Nuestra civilización posee el potencial para crear una forma de vida muy superior».

La firma de McAlear se está dedicando con mucho interés a la mezcla esencial de tecnologías biológicas y de ordenadores —un *chip* que es un millón de veces más pequeño y millones de veces más veloz que el cerebro humano; un *chip* que pueda comer, pensar e incluso reproducirse— un *biochip.*

Y para que el lector no se sienta tentado de descartar este tema como delirios de un científico desvariado, tenga en cuenta que empresas como **Gould, Westinghouse** y **General Electric** han enviado solicitudes al gobierno federal para consolidar sus investigaciones en *biochips* y «ordenadores moleculares». **Hughes Aircraft** actualmente está dedicando una pequeña fortuna a la investigación de *biochips.* **Sharp** y **Mitsubishi,** en Japón, están elaborando nuevas funciones para su investigación en *biochips*, y algo similar se está llevando a cabo en

Australia y la Unión Soviética.

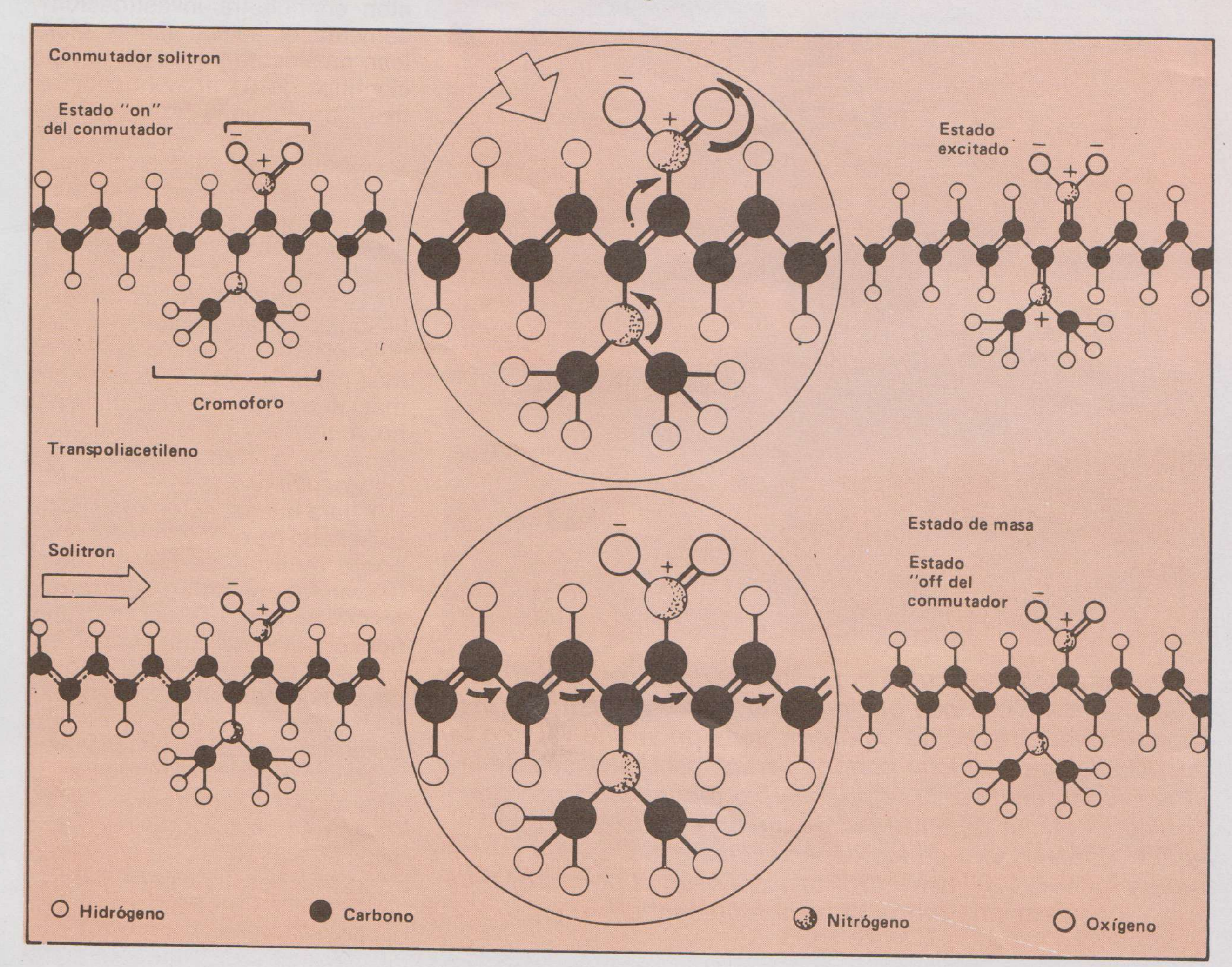
Como una célula nerviosa, un componente individual del circuito de un chip de ordenador en **estado sólido** está formado de miles de moléculas. Estas miles de moléculas individuales se combinan todas para formar un transistor en un *chip*, o una célula en un cerebro, la cual puede procesar un bit de información o ejecutar una función por vez. Pero imagine las capacidades de un ordenador que utiliza moléculas individuales como elementos de su circuito. Una partícula de material del tamaño de un transistor individual en un *chip* de ordenador podría realizar miles de funciones, una para cada molécula.

Utilizando moléculas vivientes que funcionen como componentes electrónicos, un ordenador con la potencia de los ordenadores centrales más grandes que exigen hoy, sería del tamaño de un microbio.

Las moléculas proteicas, como porphyrin compuesto, probablemente sean candidatas para los bloques de la construcción de un ordenador de *biochips*. Las moléculas proteicas tienen la capacidad de auto-congregarse en presencia de determinadas moléculas, tales como DNA (ácido desoxirrobonucleico), y a través del uso de la ingeniería genética, podrían producirse bacterias u otros organismos que, de hecho, formarán el ordenador de *biochips*, utilizando su propio código DNA como un grupo de instrucciones.

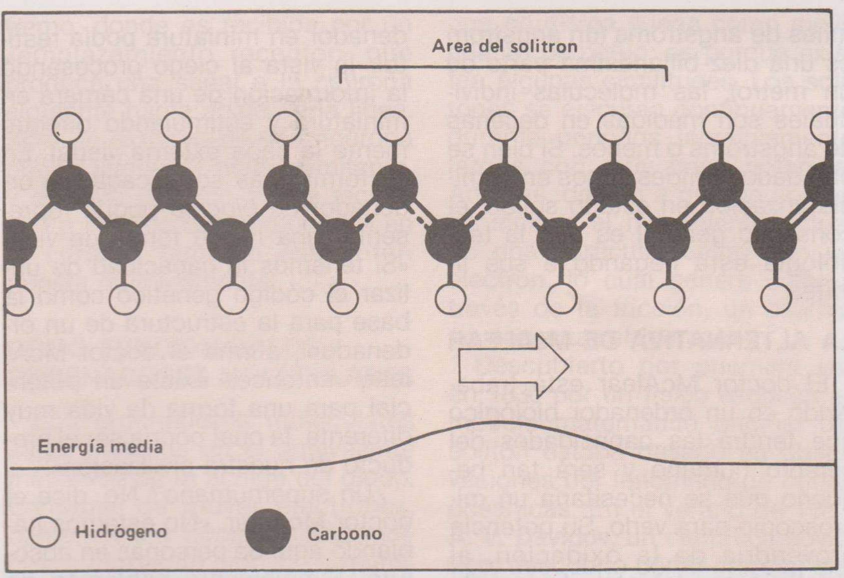
Las diferencias principales entre un cerebro humano y un ordenador de **estado sólido,** es el ta-

El conmutador Solitron se funda en la capacidad teórica de propagar de manerra que readapta los enlaces de las moléculas.
En este ejemplo aparece un conmutador molecular consistente en un tinte absorbente de luz, o cromoforo, embutido en una larga cadena de transpoliacetileno. Cuando el cromoforo está en estado *on*, absorbe luz y puede pasar de su estado bajo al excitado. La parte de la figura inclinada en la ilustración indica el mecanismo. Cada flecha representa el movimiento de una pareja de electrones. La propagación de un Solitron a través de la cadena readapta la disposición de los enlaces del cromoforo y lo conmuta al estado *off*, por lo que ya no puede ser activado por la luz.
Este mecanismo es reversible. La propagación de un segundo Solitron vuelve al cromoforo a su estado original.



maño y la complejidad. El cerebro humano tiene millones de neuronas más que transistores existen en la CPU (unidad central de proceso) del ordenador más complejo y, por supuesto, el cerebro ocupa un espacio bastante más pequeño.

Por otra parte, antes de que los ordenadores de estado sólido puedan alcanzar un nivel de miniaturización igual al del cerebro, deben resolverse serios problemas de disipación del calor y acompañamiento capacitivo e inductivo (diafonía). Es más, es imposible grabar *chips* de circuitos integrados con precisión cuando los transistores individuales en el *chip* alcanzan el tamaño de la longitud de onda de la luz o de los rayos-x. Puesto que las partes más pequeñas de los circuitos modernos en estado sólido son



Uno de los enfoques utilizados para crear cables moleculares dentro de los biochips son los polímeros eléctricamente conductores. La conductividad se debe, aunque no parece estar confirmado del todo, a los Solitrones. Esto no serían más que un tipo de perturbaciones electrónicas similares a una ola, que se propagan por una cadena de transpoliacetileno, intercambiando iones en su camino.

miles de angstroms (un angstrom es una diez billonésima parte de un metro), las moléculas individuales son medidas en decenas de angstroms o menos. Si bien se han dado grandes pasos en la miniaturización en estado sólido, el consenso general es que la tecnología está llegando a sus límites.

LA ALTERNATIVA DE McALEAR

El doctor McAlear está trabajando en un ordenador biológico que tendrá las capacidades del cerebro humano y será tan pequeño que se necesitaría un microscopio para verlo. Su potencia provendría de la oxidación, al igual que con las otras formas de vida, y lo más divertido: podría reproducirse.

En su forma primitiva, dicho ordenador en miniatura podía restituir la vista al ciego procesando la información de una cámara en miniatura y estimulando directamente la capa externa visual. En su forma más sofisticada, un ordenador de *biochip* podría representar una nueva forma de vida. «Si tenemos la capacidad de utilizar el código genético como la base para la estructura de un ordenador», afirma el doctor McAlear, «entonces existe un potencial para una forma de vida muy diferente, la cual podría ser el producto de nuestra civilización».

¿Un superhumano? No, dice el doctor McAlear. «No estamos hablando aquí de personas en absoluto. Simplemente hablamos de sistemas nuevos. Sería ideal e ingenuo suponer que dentro de un billón de años, o cuatro billones de años —el tiempo transcurrido desde el origen de la vida en la tierra— existirá algo que se asemeje a un humano.»

ORDENADOR VERSUS CEREBRO

Las funciones del ordenador y del cerebro son notablemente similares en que ambos están basados en uno de los conceptos eléctricos más simples, el efecto diodo, ísta es la tendencia de ciertas estructuras mecánicas, eléctricas o biológicas que permiten que la energía fluya en una dirección y resista su flujo en otra dirección. En un dispositivo electrónico, el flujo de la energía está formado por electrones, y un buen ejemplo de eso es un diodo de tubo electrónico, la base de la electrónica moderna.

En un diodo de tubo electrónico, un elemento llamado cátodo, es calentado a unos 1.340 grados Fahrenheit, produciéndose una nube de electrones libres. Otro elemento, llamado ánodo, es introducido en el tubo con una ligera carga positiva (una carga positiva representa una ausencia de electrones). Este atrae a los electrones libres alrededor del cátodo, produciendo una corriente eléctrica (el movimiento de electrones). Los electrones no pueden trasladarse desde el ánodo hasta el cátodo; el ánodo, estando frío es incapaz de emitir electrones.

Un diodo de estado sólido funciona de una manera muy parecida. Dos capas de materiales «semiconductores», una estimulada con una impureza que puede denotar, o abandonar electrones, y la otra estimulada para atraer electrones, están prensadas una contra la otra. Como en un diodo de tubo electrónico, estos elementos básicos son llamados el ánodo y cátodo, y permiten que los electrones fluyan en sólo una dirección.

Cuando se introduce una «compuerta» entre el ánodo y cátodo de un diodo (tubo electrónico o estado sólido) se agrega la capacidad de conectar y desconectar el flujo de electrones. Esta compuerta, que se denomina rejilla de válvula eléctrica en un tubo y base en un transistor, permite que un voltaje «de control» pequeño conecte y desconecte un voltaje mucho mayor entre el cátodo y el ánodo. También nos permite utilizar el diodo como un *switch:* está en *on* o en *off,* abierto o cerrado, como resultado del voltaje aplicado a la compuerta (la condición de *on* u *off* puede estar convenientemente representada por los dígitos 1 o 0).

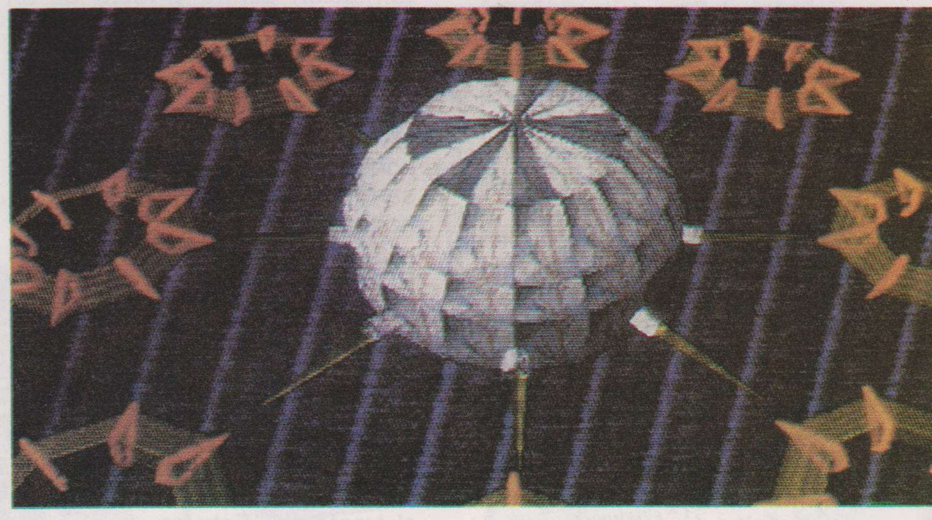
En el cerebro, las células individuales llamadas neuronas funcionan como diodos. Una química, llamada neurorremitente, es liberada en un extremo de una neurona y viaja hasta el otro extremo, donde es recibida por un «emplazamiento receptor» que produce una señal a la neurona siguiente hacia abajo de la línea. Sólo un extremo de una neurona puede liberar al **neurorremitente**, y sólo un extremo puede recibirlo. La función de «compuerta» en el cerebro también está interpuesta químicamente.

COMO FUNCIONAN LOS ORDENADORES MOLECULARES

Los ordenadores moleculares están basados en el equivalente mecánico de la acción del diodo. En lugar de mover electrones, como en los circuitos electrónicos, o moviendo productos químicos, como en una neurona, ellos operan por el movimiento de «solitons». Un soliton es virtualmente una onda de no disipación que se mueve a lo largo de la columna dorsal de una molécula un paso por vez. Los descubrimientos recientes en Estados Unidos y Alemania de moléculas, que permiten que una onda de soliton se muevan sólo en una dirección —un diodo biológico— y sea colocada en estado *on* u *off,* han generado todo tipo de especulaciones en torno al papel que los solitones pueden jugar en los ordenadores del futuro.

Si el concepto de una «onda» que no disipa suena como movimiento perpetuo, es porque existen algunas similitudes. Los solitones se ejecutan continuamente durante períodos increíblemente largos con muy poca potencia. Además, debido a que un soliton no incluye el movimiento actual de una partícula, como con un electrón, lo cual genera calor a través de la fricción, un *biochip* no necesita refrigeración.

Descubierto por priemera vez en 1830 por un físico escocés, el módulo matemático original del soliton estaba basado en observaciones del fenómeno de ondas océanicas. En el verano de 1983, **A. V. Davidov**, un científico soviético, descubrió cómo mezclar sistemas aparentemente diferentes



de matemáticas no lineal y mecánica cuántica, suministrando una base teórica para la aplicación del soliton o fenómenos moleculares.

Otro avance importantísimo para los solitones ocurrió en 1983, en California. Allí, un grupo de científicos que se reunió bajo el lema de «Electrodinámica No-Lineal en Sistemas Biológicos» demostró la viabilidad de la teoría. De hecho salió del estuche de la teoría y entró en el laboratorio», comenta McAlear.

Otros avances incluyen el descubrimiento de que los solitones

son los mediadores para la mayoría de las actividades biológicas. «Esto sucede en todos nosotros, todo el tiempo», dice McAlear. «Resulta que la mayoría de los fenómenos biológicos tienen un factor de soliton. Esto incluye fenómenos de membranas, actividad enzimática, y muchas otras acciones. Es el aspecto dinámico de las cosas vivientes que se había extraviado de nuestra comprensión.»

IMPLANTACIONES DE ORDENADOR

Uno de los problemas más grandes con el uso de los ordenadores en conjunción con el cerebro humano ha sido el tamaño grueso de cables cuando se lo compara con las delicadas neuronas individuales. Esto también presenta un problema para los *biochips*, ya que los elementos del circuito son mucho más pequeños que cualquier cable físico. Sin embargo, en 1981, McAlear dio un significativo paso adelante al solucionar este problema.

Su investigación se relacionó con un descubrimiento en la década de los 50, en que la gente ciega, cuando ciertas partes de los cerebros fueran simuladas por electricidad durante la cirugía de cráneo abierto, podía ver amplios resplandores brillantes de luz, llamados fosfenos. La razón de que los fosfenos fueran tan amplios y tan brillantes es que los cables relativamente grandes utilizados para introducir la corriente dentro del cerebro estaban estimulando a los grupos de células nerviosas en lugar de neuronas individuales. Aunque el proceso fue refinado hasta el punto en que podrían ser vistas las formas primitivas, éstas parecían estar formadas por «puntos» que tenían el aspecto de haber sido producidos por un borrador de lápiz sumergido en pintura blanca.

Para estimular las células nerviosas individuales e incrementar la claridad de las imágenes, McAlear desarrolló una capa aislante de proteína a través de la cual miles de orificios son perforados con un revolver de electrones. Luego se aplican los electrodos en la superficie externa de la proteína, y se estimula que las células nerviosas crezcan a través de los agujeros en la proteína y realicen un contacto selectivo con los electrodos. Esta técnica permite el *input* directamente al cerebro desde una cámara controlada por ordenador. Podría utilizarse un sistema similar para conectar *biochips* a los cables, según McAlear, quien ha patentado su invención y espera tener el dispositivo en el mercado en los próximos años.

Otras técnicas para obtener información dentro y fuera de un *biochip* incluyen «cables» semiconductores más grandes que crecerán progresivamente desde puntos estratégicos en el *chip*, y el uso de sustancias biológicas conductoras, las cuales son desarrolladas dentro de las conexiones con los cables.

BENEFICIOS MOLECULARES

De todos los factores de limitación para la tecnología actual de silicona, la diafonía es uno de los más formidables. Usted probablemente habrá tenido una llamada telefónica de larga distancia y ha escuchado otra conversación en el trasfondo. Este es un ejemplo de diafonía. Es el resultado de la tendencia de mover electrones para crear campos electromagnéticos, los cuales son entonces irradiados dentro de cables contiguos —como los haces de larga distancia que utiliza la compañía de teléfonos. Este fenómeno hace posible los generadores eléctricos, los motores y los transmitores de radio y TV y receptores. También provoca interferencias en las comunicaciones de larga distancia y puede confundir totalmente el multicircuito del ordenador.

Cuando los transistores individuales y las sendas de los electrones en un *chip* de ordenador son tan pequeños y están tan próximos que uno funciona a través del «campo» irradiado por otro, se produce la diafonía. Conocida en los círculos de diseño como acomplamiento capacitivo e inductivo, la diafonía puede hacer que un *chip* funcione mal, provocando toda clase de errores de datos, y es un obstáculo importante para la miniaturización ulterior de los circuitos integrados.

Los *biochips*, por otro lado, no tienen este problema porque no utilizan electrones y no producen ninguna radiación electromagnética. No obstante, los solitones interactúan con luz y, en algunos materiales, pueden, de hecho, producir luz; pero según McAlear, eso es relativamente sencillo contra lo cual protegerse.

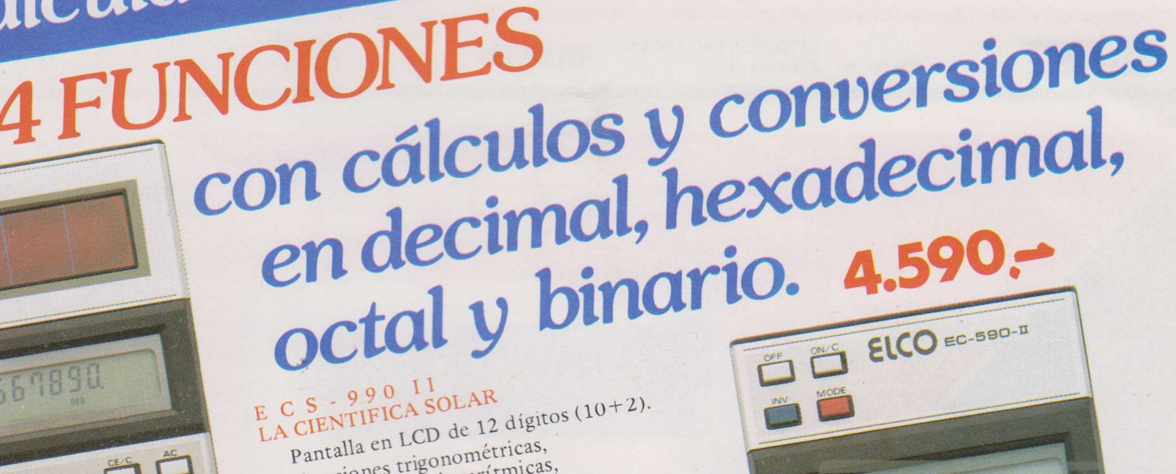
SUPERVIVENCIA DEL ORDENADOR

El fenómeno de radiación electromagnética también ha despertado el interés en el Pentágono. Según **Ike Skelton,** miembro del Congreso Norteamericano, los EE.UU. y la Unión Soviética están considerando una explosión nuclear de pulso electromagnético en el espacio que pudiera enviar una carga eléctrica extraordinaria a los objetos de metal —incluyendo todos los circuitos integrados y máquinas fabricadas de metal— en un hemisferio entero, volviéndolos inútiles. El ordenador molecular de McAlear, sin embargo, podría sobrevivir a dichas cargas y continuar operando.

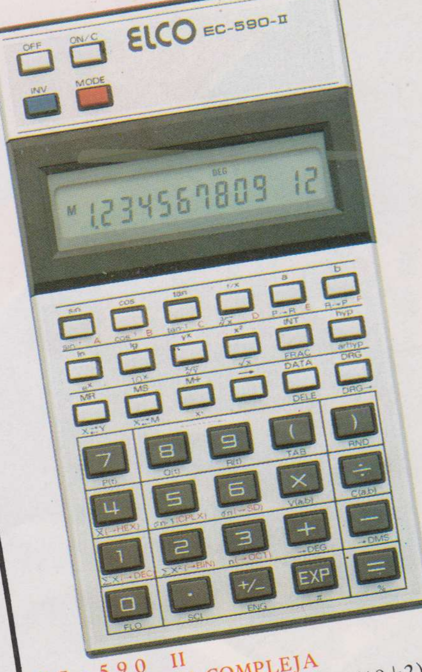
Los problemas teóricos que rodean a los solitones y a la ingeniería genética han sido todos resueltos, y tal vez surja una nueva generación de ordenadores de un laboratorio biológico. El único sobreviviente del holocausto nuclear ciertamente puede ser la nueva forma de vida con un cerebro de ordenador molecular de McAlear.

ELCO calculadoras para estudiantes:

94 FUNCIONES



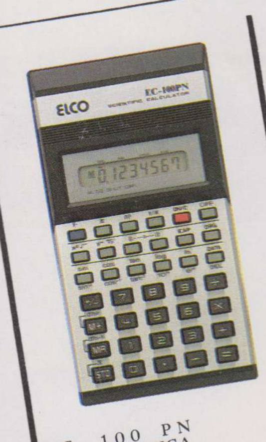
4.590.-



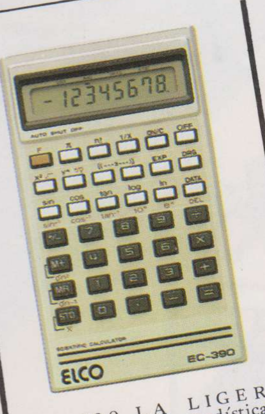
E C S - 9 9 0 I I
LA CIENTIFICA SOLAR

Pantalla en LCD de 12 dígitos (10+2).
Funciones trigonométricas, exponenciales, logarítmicas, estadística e hiperbólicas y sus inversas.
Conversiones de grados centesimales a sexagesimales y de coordenadas rectangulares a polares y viceversa.
15 niveles de paréntesis.
Notaciones científicas, ingenieril o con selector de décimales.
Celdas solares de alta resolución.

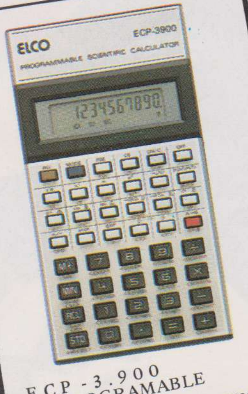
5.590.-



E C - 1 0 0 P N
LA ECONOMICA
31 funciones con estadística y 8 dígitos.
Usa dos pilas normales.
2.990 ptas.



E C - 3 9 0 L A L I G E R A
31 Funciones con estadísticas y 8 dígitos.
Apagado automático.
3.290 ptas.



E C P - 3 . 9 0 0
LA PROGRAMABLE
Admite dos programas y 45 pasos de programación en memoria constante.
Con toma de decisiones.
64 funciones científicas y 10 dígitos.
6.590 ptas.

E C - 5 9 0 I I
LA CIENTIFICA COMPLEJA
— Pantalla en LCD de 12 dígitos (10+2).
Funciones trigonométricas, exponenciales, logarítmicas, hiperbólicas y sus inversas.
Conversiones de grados centesimales a sexagesimales de coordenadas rectangulares a polares.
Funciones estadísticas: N, x, x^2, s, σ, DATA, CD, CAD, %.
Notaciones científicas, ingenieril o con el número de decimales deseado en pantalla.

ALVARO SOBRINO

ELCO-1 Electrónica de Consumo-1, S.A.
c/ Rufino González, 6
Telfs.: 204 76 56 y 204 05 70 - Telex 42489 ELCO E
28037 MADRID